

Der Entwicklungsaufwand der Anwendungsintegration:

Eine empirische Untersuchung der Einflussfaktoren

Inauguraldissertation zur Erlangung des Doktorgrades
der Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Universität zu Köln

2006

Vorgelegt von

Dipl.-Kfm. Holger Wagner
aus Bonn

Referent: Prof. Dr. W. Mellis

Korreferent: Prof. Dr. D. Seibt

Promotionsdatum: 12.06.2006

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VIII
Abkürzungsverzeichnis	XI

1 Einleitung	1
1.1 Analyse der Problemstellung.....	1
1.1.1 Praxisproblem.....	2
1.1.2 Forschungsfrage	4
1.1.3 Forschungsproblem	5
1.2 Erkenntnisziel der Arbeit.....	9
1.3 Wissenschaftstheoretische Einordnung der Arbeit.....	10
1.4 Begriffliche Grundlagen.....	12
1.4.1 Herleitung einer Definition der Anwendungsintegration.....	13
1.4.2 Abgrenzung der Anwendungsintegration gegenüber anderen Integrationsansätzen	15
1.4.3 Anwendungsintegration als Ausprägung der Softwareentwicklung	17
1.4.4 Entwicklungsaufwand und Aufwandschätzung	18
1.5 Aufbau der Arbeit.....	21
2 Konzeptioneller Bezugsrahmen der Untersuchung	22
2.1 Integrationsprobleme.....	24
2.1.1 Anwendungssysteme als Abbildungen von Ausschnitten der betrieblichen Realität.....	24
2.1.2 Abhängigkeiten zwischen den Anwendungssystemen.....	28
2.1.3 Hintergrund eines Integrationsbedarfs	31
2.1.4 Dimensionen eines Integrationszustands.....	34
2.1.5 Integrationsprobleme als unzureichende Integrationszustände von Anwendungssystemen	38

2.1.6	Charakteristika der Anwendungssysteme	40
2.1.6.1	Konzeptionelle und technische Merkmale der Anwendungssysteme	40
2.1.6.2	Zentrale Herausforderungen für die Anwendungsintegration.....	47
2.2	Integrationslösungen	48
2.2.1	Integrationslösungen als einzigartige Softwaresysteme.....	48
2.2.2	Kategorien von Integrationslösungen.....	50
2.2.3	Gestaltungsmerkmale einer Integrationslösung	52
2.2.3.1	Automatisierte Interaktionen als Zweck von Integrationslösungen	53
2.2.3.2	Funktionsbezogene Gestaltungsmerkmale	59
2.2.3.3	Qualitätsbezogene Gestaltungsmerkmale.....	62
2.2.4	Integrationsarchitekturen.....	65
2.2.4.1	Gestaltungsmuster und Middleware	65
2.2.4.2	Serviceorientierung	71
2.2.4.3	Orientierung an Standards	73
2.2.4.4	Einsatz von Integrationsprodukten.....	76
2.3	Integrationsvorhaben	80
2.3.1	Generisches Modell von Entwicklungsvorhaben	81
2.3.2	Besonderheiten der Integrationsvorhaben	84
2.3.3	Erfolgsfaktoren und Risiken für Integrationsvorhaben	92
2.3.3.1	Einflussfaktoren auf den Erfolg der Anwendungsintegration.....	92
2.3.3.2	Risiken für Integrationsvorhaben	96
2.4	Entwicklungsaufwand und die ihn beeinflussenden Faktoren	98
2.4.1	Grundlegende Überlegungen zum Entwicklungsaufwand	98
2.4.2	Die kontinuierliche Suche nach den Determinanten des Entwicklungsaufwands.....	100

3	Stand der Erkenntnis zu den Einflussfaktoren auf den Entwicklungsaufwand.....	102
3.1	Erkenntnisse zur Neuentwicklung von Anwendungssystemen.....	103
3.1.1	Der Umfang der zu entwickelnden Software als Arbeitslast	103
3.1.2	Weitere Einflussfaktoren auf den Entwicklungsaufwand	107
3.2	Erkenntnisse zum Einsatz von Standardsoftwarekomponenten.....	115
3.3	Erkenntnisse zur Entwicklung eines Systems von Softwaresystemen.....	120
3.4	Erkenntnisse zum integrationsorientierten Reengineering von Altsystemen.....	124
3.5	Zusammenfassende Beurteilung der Erkenntnisse.....	129
4	Konzeption der Untersuchung	131
4.1	Definition der Untersuchungsziele.....	131
4.2	Theoretische Grundlagen der Untersuchung.....	132
4.3	Spezifizierung der Untersuchungsmethode.....	134
4.3.1	Repertory Grid Technique.....	135
4.3.2	Ausgestaltung der Methode zur Erhebung persönlicher Konstrukte....	136
4.3.2.1	Zusammenfassende Darstellung der Methode	136
4.3.2.2	Bestimmung der betrachteten Elemente.....	137
4.3.2.3	Gestaltung der Vergleiche	138
4.3.2.4	Weiterführendes Verständnis der Konstruktsysteme	142
4.3.3	Eignung der Methode zur Erreichung des Erkenntnisziels der Arbeit.	143
4.3.4	Gestaltung des Auswahlverfahren der Auskunftspersonen.....	145
4.3.5	Anforderungen an die Auswahl der Integrationsvorhaben.....	149
4.3.6	Präzisierung des Bezugsrahmens für die Vergleiche	150
4.4	Ablauf der Untersuchung	152
4.4.1	Vorbereitung der Interviews.....	152
4.4.2	Identifikation der Untersuchungsteilnehmer	153
4.4.3	Durchführung der Interviews	154
4.4.3.1	Einführung und Vereinbarung zur Gesprächsaufzeichnung	155
4.4.3.2	Schaffung begrifflicher Grundlagen.....	155
4.4.3.3	Erfahrungshintergrund der Untersuchungsteilnehmer	157

4.4.3.4	Erhebung der persönlichen Konstrukte	157
4.4.3.5	Kommentierung der erhobenen Konstrukte	162
4.4.3.6	Kommentierung der betrachteten Integrationsvorhaben	163
4.4.3.7	Rückmeldung der Untersuchungsteilnehmer	165
4.4.4	Transkription der Gesprächsaufzeichnungen	166
5	Ergebnisse der Untersuchung	168
5.1	Charakterisierung der Untersuchungsteilnehmer	168
5.2	Charakterisierung der betrachteten Integrationsvorhaben.....	173
5.3	Vorgehen zur inhaltlichen Analyse der erhobenen Konstrukte.....	182
5.3.1	Bestimmung der Häufigkeiten der Konstrukte.....	182
5.3.2	Begriffliche Harmonisierung der Konstrukte.....	184
5.3.3	Einteilung der erhobenen Konstrukte in Gruppen und Themenbereiche	185
5.3.4	Trendanalyse der inhaltlichen Gemeinsamkeiten der erhobenen Konstrukte	188
5.4	Ergebnisse der inhaltlichen Analyse der erhobenen Konstrukte.....	189
5.4.1	Kommentierung der Konstrukte durch die Untersuchungsteilnehmer.	190
5.4.2	Die abgegrenzten Themenbereiche im Überblick	192
5.4.3	Einflussfaktoren im Zusammenhang mit den Anforderungen an die Integrationslösung	196
5.4.4	Einflussfaktoren im Zusammenhang mit den Charakteristika der zu integrierenden Anwendungssysteme	203
5.4.5	Einflussfaktoren im Zusammenhang mit der Gestaltung der Integrationslösung	208
5.4.6	Einflussfaktoren im Zusammenhang mit den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen .	214
5.4.7	Einflussfaktoren im Zusammenhang mit dem Testen der Integrationslösung und der Überleitung in die Nutzung	218
5.4.8	Einflussfaktoren im Zusammenhang mit dem Kunden und den Stakeholdern der Integrationslösung	220
5.4.9	Einflussfaktoren im Zusammenhang mit den Projektmitarbeitern.....	224

5.4.10	Einflussfaktoren im Zusammenhang mit dem Management des Integrationsvorhabens	228
5.4.11	Einflussfaktoren im Zusammenhang mit den Rahmenbedingungen des Integrationsvorhabens	231
5.5	Die Untersuchungsergebnisse im Kontext bisheriger Erkenntnisse.....	236
6	Beurteilung der Untersuchung und Schlussfolgerungen	244
6.1	Beurteilung der Untersuchung.....	244
6.1.1	Angemessenheit der Untersuchung	244
6.1.2	Intersubjektive Nachvollziehbarkeit der Untersuchung	245
6.1.3	Konsistenz der Untersuchungsergebnisse	247
6.1.4	Einschränkungen der Untersuchungsergebnisse	248
6.2	Wissenschaftliche Relevanz der Untersuchungsergebnisse	250
6.2.1	Erkenntnisbeitrag der Untersuchung	250
6.2.2	Perspektiven für die wissenschaftliche Verwendung der Untersuchungsergebnisse	252
6.2.2.1	Priorisierung der Konstrukte und Untersuchung der Wirkungszusammenhänge.....	252
6.2.2.2	Präzisierung der Konstrukte	255
6.3	Praktische Relevanz der Untersuchungsergebnisse	263
7	Fazit	267
	Literaturverzeichnis	268
	Anhang A: Detailinformationen zu den erhobenen Konstrukten	292
	Anhang B: Eine Checkliste für die Aufwandschätzung von Integrationsvorhaben	318

Abbildungsverzeichnis

Abbildung 1-1:	Aufbau der Arbeit.....	22
Abbildung 2-1:	Zyklus von Integrationsproblem, Integrationsvorhaben und Integrationslösung	23
Abbildung 2-2:	Richtung der Abhängigkeiten zwischen Anwendungssystemen	31
Abbildung 2-3:	Integrationsproblem als fehlende Kongruenz von Integrationsbedarf und Integrationszustand.....	39
Abbildung 2-4:	Einschicht-, Zweischicht- und Dreischichtarchitekturen.....	45
Abbildung 2-5:	Gestaltungsmerkmale einer Integrationslösung.....	53
Abbildung 2-6:	Alternative Interaktionsschichten	55
Abbildung 2-7:	Magic Quadrant for Application Integration Suites, 2. Quartal 2004	78
Abbildung 2-8:	„Giga-Scorecard“ zur Beurteilung von Integrationsprodukten	79
Abbildung 2-9:	Generisches Modell von Entwicklungsvorhaben	84
Abbildung 4-1:	Darstellung der Konstruktkarte zur Erläuterung der Untersuchungsmethode gegenüber den Auskunftspersonen	158
Abbildung 4-2:	Darstellung einer unbeschrifteten Elementkarte.....	159
Abbildung 4-3:	Darstellung einer unbeschrifteten Konstruktkarte	161
Abbildung 4-4:	Darstellung der Rückseite einer unbeschrifteten Elementkarte	165
Abbildung 5-1:	Berufserfahrung der Untersuchungsteilnehmer in Jahren	169
Abbildung 5-2:	Anzahl der Integrationsvorhaben, an denen die Untersuchungsteilnehmer mitgewirkt haben	170
Abbildung 5-3:	Anzahl der Integrationsvorhaben, an deren Planung die Untersuchungsteilnehmer mitgewirkt haben	171
Abbildung 5-4:	Variation der eingesetzten Technologien	173
Abbildung 5-5:	Dauer der in der Untersuchung betrachteten Integrationsvorhaben	174

Abbildung 5-6:	Anzahl der betrachteten Integrationsvorhaben, die in einem bestimmten Jahr abgeschlossen wurden	175
Abbildung 5-7:	Zuständigkeit der Untersuchungsteilnehmer in den betrachteten Integrationsvorhaben.....	176
Abbildung 5-8:	Beurteilung der Informationsorientierung der Integrationsvorhaben	177
Abbildung 5-9:	Beurteilung der Funktions- bzw. Serviceorientierung der Integrationsvorhaben	177
Abbildung 5-10:	Beurteilung der Geschäftsprozessorientierung der Integrationsvorhaben	178
Abbildung 5-11:	Integrationsreichweite in den betrachteten Integrationsvorhaben	179
Abbildung 5-12:	Internationalität der betrachteten Integrationsvorhaben	179
Abbildung 5-13:	Die in den betrachteten Integrationsvorhaben zu integrierenden Anwendungssysteme	180
Abbildung 5-14:	Neuartigkeit der Integrationslösungen in den betrachteten Integrationsvorhaben	181
Abbildung 5-15:	Kumulierter Anteil der ermittelten Gruppen in Abhängigkeit vom kumulierten Anteil der durchgeführten Interviews	189

Tabellenverzeichnis

Tabelle 2-1:	Beispielhafte Wirkungsarten operativer Nutzeffekte der Integration von Anwendungssystemen.....	33
Tabelle 2-2:	Beispielhafte Qualitätsmerkmale zur Charakterisierung von Integrationszuständen	38
Tabelle 2-3:	Beispiele für Eigenschaften externer Schnittstellen der Anwendungssysteme	43
Tabelle 2-4:	Ausgewählte Gestaltungsmuster für Integrationsarchitekturen	69
Tabelle 2-5:	Ausgewählte Kategorien von Middleware	70
Tabelle 2-6:	Merkmale von Standards für die Gestaltung von Integrationsarchitekturen	75
Tabelle 2-7:	Sichten zur Beschreibung von Integrationsarchitekturen gemäß Ruh, Maginnes und Brown	88
Tabelle 2-8:	Beispiele für Rollen in Integrationsvorhaben nach Yee und Apte	91
Tabelle 2-9:	Komponenten der Erfolgsvoraussetzungen der Anwendungsintegration gemäß Klesse, Wortmann und Schelp	95
Tabelle 2-10:	Risikobereiche von Integrationsvorhaben gemäß Lam	98
Tabelle 3-1:	Kategorien und Klassifizierung von Funktionen zum Zählen von Function Points	106
Tabelle 3-2:	Einflussfaktoren auf die Produktivität gemäß Walston und Felix	110
Tabelle 3-3:	Einflussfaktoren auf den Entwicklungsaufwand im COCOMO II-Verfahren	115
Tabelle 3-4:	Einflussfaktoren auf den Aufwand für die Entwicklung des Programmcodes für die Integration von Standardsoftwarekomponenten gemäß dem COCOTS-Verfahren	119

Tabelle 3-5:	Einflussfaktoren auf den Aufwand für die Entwicklung von SoS gemäß des Post-Architecture Model des COSOSIMO-Verfahrens	122
Tabelle 3-6:	Einflussfaktoren auf den Aufwand für die Entwicklung von SoS gemäß des Early-Design Model des COSOSIMO-Verfahrens	124
Tabelle 3-7:	Einflussfaktoren auf den Aufwand für das integrationsorientierte Reengineering von Altsystemen gemäß Jung	127
Tabelle 4-1:	Theoretisches versus statistisches Sampling	148
Tabelle 5-1:	Branchen, in denen die Auskunftspersonen Integrationsvorhaben durchgeführt haben	172
Tabelle 5-2:	Kumulierte Anzahl der ermittelten Gruppen in Abhängigkeit von der Anzahl der durchgeführten Interviews	188
Tabelle 5-3:	Beurteilung der Bedeutung der erhobenen Konstrukte für den Entwicklungsaufwand.....	190
Tabelle 5-4:	Unterschiedene Themenbereiche der erhobenen persönlichen Konstrukte	193
Tabelle 5-5:	Kommentierung der Konstrukte in den unterschiedenen Themenbereichen.....	195
Tabelle 5-6:	Gruppen der Konstrukte zu den Anforderungen an die Integrationslösung	198
Tabelle 5-7:	Persönliche Konstrukte zu den Anforderungen an die Integrationslösung	202
Tabelle 5-8:	Gruppen der Konstrukte zu den Charakteristika der zu integrierenden Anwendungssysteme	204
Tabelle 5-9:	Persönliche Konstrukte zu den Charakteristika der zu integrierenden Anwendungssysteme	207
Tabelle 5-10:	Gruppen der Konstrukte zur Gestaltung der Integrationslösung	209
Tabelle 5-11:	Konstrukte zur Gestaltung der Integrationslösung	213

Tabelle 5-12:	Gruppen der Konstrukte zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen	215
Tabelle 5-13:	Konstrukte zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen	217
Tabelle 5-14:	Gruppen der persönlichen Konstrukte zum Testen der Integrationslösung und zur Überleitung in die Nutzung	218
Tabelle 5-15:	Konstrukte zum Testen der Integrationslösung und zur Überleitung in die Nutzung	219
Tabelle 5-16:	Gruppen der Konstrukte zum Kunden und den Stakeholdern der Integrationslösung	221
Tabelle 5-17:	Persönliche Konstrukte zum Kunde und den Stakeholdern des Integrationsvorhabens	224
Tabelle 5-18:	Die Gruppen der Konstrukte zu den Projektmitarbeitern	225
Tabelle 5-19:	Konstrukte zu den Projektmitarbeitern	228
Tabelle 5-20:	Gruppen der persönlichen Konstrukte zum Management des Integrationsvorhabens	229
Tabelle 5-21:	Konstrukte zum Management des Integrationsvorhabens	231
Tabelle 5-22:	Gruppen der Konstrukte zu den Rahmenbedingungen des Integrationsvorhabens	232
Tabelle 5-23:	Persönliche Konstrukte zu den Rahmenbedingungen des Integrationsvorhabens	235
Tabelle 5-24:	Vergleich der Gruppen der erhobenen Konstrukte mit bestehenden Erkenntnissen zum Entwicklungsaufwand von Softwareentwicklungsvorhaben	241
Tabelle 6-1:	Hinweise auf die Klassifizierung der Attribute in einigen der erhobenen Konstrukte	262
Tabelle 6-2:	Phasen und Aktivitäten der Aufwandschätzung gemäß Jorgensen und Molokken	265

Abkürzungsverzeichnis

API	Application Programming Interface
ARIS	Architektur integrierter Informationssysteme
B2C	Business-to-Customer
BML	Business Model Language
BPEL	Business Process Execution Language
BPR	Business Process Reengineering
CMM	Capability Maturity Model
COCOMO	Constructive Cost Model
COCOTS	Constructive COTS Integration Cost Model
CORBA	Common Object Request Broker Architecture
COSOSIMO	Constructive System of Systems Integration Model
COTS	Commercial-off-the-shelf
CSE	Center for Software Engineering
DCOM	Distributed Component Object Model
EAI	Enterprise Application Integration
engl.	im Englischen
ERP-System	Enterprise Resource Planning System
ETL	Extract-Transform-Load
IFPUG	International Function Point Users Group
IT	Informationstechnik
IVAS	Integriertes verteiltes Anwendungssystem
MOM	Message-oriented Middleware
RMI	Remote Method Invocation
ROI	Return on Investment
RPC	Remote Procedure Call
SCM	Supply Chain Management
SOA	Serviceorientierte Architektur
SOAP	Simple Object Access Protocol

SoS	System of Systems
USC	University of Southern California
WSDL	Web-Service Description Language
XML	Extensible Markup Language

„The dynamism of the software field means that the software estimation discipline needs to be continually reinventing itself.”¹

1 Einleitung

1.1 Analyse der Problemstellung

Gemäß Booth, Colomb und Williams gründet eine wissenschaftliche Arbeit auf einem Praxisproblem, d. h. einem unbefriedigendem Zustand in der Praxis, dessen Fortbestand mit Kosten bzw. nicht realisierten Nutzenpotenzialen verbunden ist.² Sie führen weiterhin aus, dass aus einem Praxisproblem verschiedene Fragestellungen abgeleitet werden können, die beantwortet werden müssen, um es zu lösen. Eine davon definiert als Forschungsfrage die Perspektive der wissenschaftlichen Auseinandersetzung mit dem Praxisproblem. Die Suche nach einer Antwort zeigt oftmals eine unzureichende Erkenntnis hinsichtlich bestimmter Sachverhalte auf, d. h. sie definiert ein Forschungsproblem. Der Beitrag zur Lösung dieses Forschungsproblems, zur Beantwortung der Forschungsfrage und zur Lösung des Praxisproblems definieren den Gegenstand einer wissenschaftlichen Arbeit.

Die vorliegende Arbeit widmet sich dem Praxisproblem, dass keine geeigneten Methoden zur Schätzung des Entwicklungsaufwands der Anwendungsintegration verfügbar sind, infolge dessen viele Integrationsvorhaben durch Termin- und Kostenüberschreitungen gekennzeichnet sind. Daraus wird die Forschungsfrage abgeleitet, welche Faktoren in die Aufwandschätzung der Anwendungsintegration einbezogen werden müssen, um zuverlässige Schätzergebnisse zu erzielen. Zur Beantwortung dieser Frage erscheint zunächst die Lösung des Forschungsproblems erforderlich, dass keine angemessenen empirischen oder theoretischen Erkenntnisse vorliegen, auf deren Grundlage Schlüsse auf die maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration gezogen werden könnten. Die vorliegende Arbeit soll einen Beitrag zur Behebung dieses Mangels leisten und damit wichtige Voraussetzungen für eine Beantwortung der Forschungsfrage und die Lösung des Praxisproblems schaffen.

¹ Boehm, Fairley /Estimation/ 24.

² Zur Analyse der Problemstellung einer wissenschaftlichen Arbeit gemäß Booth, Colomb und Williams vgl. im Weiteren Booth, Colomb, Williams /Research/ 48-63.

1.1.1 Praxisproblem

Der Einsatz von betrieblichen Anwendungssystemen ist von wesentlicher Bedeutung für den wirtschaftlichen Erfolg der meisten Unternehmen geworden. Diese setzen oftmals eine Vielzahl verschiedenartiger Anwendungssysteme ein, die jeweils bestimmte Ausschnitte der betrieblichen Tätigkeit unterstützen.³ Zwischen diesen Ausschnitten bestehen jedoch häufig Zusammenhänge oder sie überschneiden sich in unterschiedlichem Ausmaß, woraus logische Interdependenzen zwischen den Anwendungssystemen resultieren. Werden diese dennoch isoliert von einander betrieben, so hat dies oftmals arbeitsintensive Mehrfacherfassungen von Daten, eine mangelnde Konsistenz der interdependenten Anwendungssysteme, Medienbrüche und unnötige Arbeitsunterbrechungen in den Arbeitsabläufen sowie einen hohen Einarbeitungsaufwand der Benutzer zur Folge.⁴

Seit einiger Zeit wird es daher als eines der wichtigsten Anliegen der Unternehmen genannt, die interdependenten Anwendungssysteme zusammenzuführen, d. h. sie zu integrieren.⁵ Laut Analysten-Angaben der letzten Jahre wird bereits mehr als ein Drittel aller Budgets für Informations- und Kommunikationstechnik zu diesem Zweck ausgegeben.⁶ Aktuelle Studien weisen darauf hin, dass sich dieser Trend zur Integration auch in den kommenden Jahren fortsetzen wird.⁷ Aus diesem umfangreichen Nachfragepotenzial

³ Nach einer Erhebung der META Group setzen die weltweit 2000 größten Unternehmen im Durchschnitt 49 unternehmensweite Anwendungssysteme zur Unterstützung ihrer Geschäftsprozesse ein (vgl. Renk /Return-on-Investment/ 63).

⁴ Vgl. Kurbel, Rautenstrauch /Integration Engineering/ 168.

⁵ Vgl. beispielsweise Quack /CIO/ 16; Quack /IT-Trends 2002/ 37 oder Ließmann /Schnittstellenorientierung/ 57.

⁶ Linthicum zitiert eine Studie von Forrester Research, nach der mehr als 30 Prozent der Ausgaben für Informations- und Kommunikationstechnik auf die Verknüpfung von Anwendungssystemen entfallen (vgl. Linthicum /EAI/ xvii). Alexander weist zudem auf eine Studie der Gartner Group hin, die ergab, dass die befragten Betriebe mehr als 35 Prozent der Budgets für Informations- und Kommunikationstechnik für die Integration von Anwendungssystemen ausgaben (Alexander /EAI/ 9). Einige Autoren äußern gar die Vermutung, dass zukünftig bis zu 70 % aller Ausgaben für Informations- und Kommunikationstechnik für Aktivitäten getätigt werden könnten, die mit der Integration von Anwendungssystemen in Zusammenhang stehen (vgl. Gorton, Thurman, Thomson /Application Integration/ 1).

⁷ Beispielsweise geht die Meta Group von einer weiterhin wachsenden Nachfrage nach Integrationslösungen aus (vgl. META Group (Hrsg.) /META Trends 2004-05/ 1f.).

resultiert ein dynamisch wachsender Marktsektor für Integrationstechnologien und -dienstleistungen.⁸

Innerhalb der vergangenen 10 Jahre hat sich ein Ansatz zur Lösung der Integrationsproblematik etabliert, der als Anwendungsintegration (engl. application integration) bezeichnet wird. Dieser steht für die nachträgliche Verknüpfung von Anwendungssystemen eines oder mehrerer Unternehmen, um automatisierte Interaktionen der Anwendungssysteme zur systemübergreifenden Unterstützung von Geschäftsprozessen zu realisieren. Ändernde oder erweiternde Eingriffe in die betroffenen Anwendungssysteme werden bei der Anwendungsintegration weitgehend vermieden. Insbesondere verbleiben die integrierten Anwendungssysteme selbständig lauffähig.⁹

Viele Unternehmen haben die Erfahrung gemacht, dass sich die Anwendungsintegration als deutlich aufwändiger und damit als langwieriger und teurer herausgestellt hat, als anfänglich vermutet. Erfahrene Praktiker führen die Unterschätzung des mit der Anwendungsintegration verbundenen Entwicklungsaufwands als einen der wichtigsten Gründe für das Scheitern von Integrationsvorhaben an.¹⁰ Die Schätzung des Entwicklungsaufwands ohne die Anwendung geeigneter Methoden wird als wesentliche Ursache dafür angesehen, dass die geplanten Aktivitäten meist deutlich länger dauern als vorgesehen und zudem häufig unvorhergesehene Belastungen im Projektverlauf auftreten.¹¹ Die Gefahren von mangelhaften Aufwandsschätzungen für die Softwareentwicklung sind schon seit langem erkannt. Sie werden als eine der wichtigsten Ursachen für Termin- und Kostenüberschreitungen oder gar den vorzeitigen Abbruch von Softwareentwicklungsvorhaben genannt.¹² Darüber hinaus gelten sie als wesentliche Quelle für das Auftreten von Zeitdruck während der Entwicklung, woraus oftmals qualitative Mängel der entwickelten Softwaresysteme resultieren.¹³

⁸ Verschiedene Analysten gehen für die nächsten Jahre von einem starken Wachstum des Marktes für Integrationstechnologien aus (vgl. z. B. Boston Corporate Finance /Enterprise Integration/2f.).

⁹ Vgl. Themistocleous, Irani /Taxonomy for Application Integration/ 156; Linthicum /EAI/ 3; Linthicum /Application Integration/ 1; Ruh, Maginnis, Brown /EAI/ 2; Wrazen /EAI/ 3.

¹⁰ Vgl. Lublinsky, Farrell /Why EAI Fails/ 41.

¹¹ Vgl. Lam /Enterprise Integration/ 304f.

¹² Vgl. Jones /Software Costs/ 3f.

¹³ Vgl. Costello /Deadline/ 15-19.

Die existierenden Methoden zur Schätzung des Entwicklungsaufwands für Softwaresysteme fokussieren überwiegend auf die Programmierung der Systeme. Diese Tätigkeit wird traditionell als der arbeitsintensivste Teil der Softwareentwicklung angesehen. Im Mittelpunkt der Betrachtung stehen daher oftmals der Umfang des zu entwickelnden Programmcodes oder der zu entwickelnden Funktionen sowie die Beurteilung verschiedener Faktoren, welche die Schwierigkeit der Programmierung determinieren sollen.¹⁴

Mit der Anwendungsintegration ist jedoch ein neuartiger Typ von Softwareentwicklungsvorhaben verbunden, für den diese Annahmen nicht zu gelten scheinen. Integrationsvorhaben zeichnen sich durch charakteristische Problemstellungen und Problemlösungen aus. Sie sind mit dem Einsatz neuer Technologien und spezieller Standardsoftwaresysteme verbunden. In ihnen vereinen sich Merkmale aus verschiedenen Teildisziplinen der Softwareentwicklung, wie beispielsweise der Neuentwicklung von Softwaresystemen, der Einführung und Anpassung von Standardsoftware sowie dem Reverse Engineering. Darüber hinaus bedeutet die Anwendungsintegration vielfach nicht nur, dass verschiedene Anwendungssysteme miteinander verknüpft werden. Es gilt auch organisatorische Grenzen von Abteilungen, Teilgesellschaften oder Unternehmen zu überwinden.¹⁵

Geeignete Methoden zur Schätzung des Entwicklungsaufwands der Anwendungsintegration sind bislang nicht bekannt geworden. Die bisherigen Erkenntnisse aus verschiedenen Teildisziplinen der Softwareentwicklung erscheinen nicht oder nur eingeschränkt auf die Anwendungsintegration übertragbar. Diese unzureichende methodische Unterstützung der Aufwandschätzung stellt angesichts der großen und weiterhin zunehmenden Bedeutung der Anwendungsintegration ein ernstzunehmendes praktisches Problem dar.

1.1.2 Forschungsfrage

Die Schätzung des Entwicklungsaufwands basiert im Wesentlichen auf zwei Schritten. Zunächst werden die konkreten Ausprägungen relevanter Einflussfaktoren bestimmt, d. h. sie werden gemessen. Aufgrund der Erfahrungen aus früheren Entwicklungsvorhaben werden darauf hin Analogieschlüsse auf den erforderlichen Entwicklungsaufwand

¹⁴ Vgl. Francalanci /Implementation Effort/ 33.

¹⁵ Diese Merkmale der Integrationsvorhaben werden im Verlauf der Arbeit näher erläutert.

gezogen.¹⁶ Die Methoden der Aufwandschätzung unterscheiden sich darin, auf welche Einflussfaktoren sie sich beziehen, wie diese gemessen werden und auf welche Weise die Analogieschlüsse erfolgen.

Soll das aufgezeigte Problem der unzureichenden methodischen Unterstützung der Aufwandschätzung für die Anwendungsintegration gelöst werden, so gilt es zunächst die Frage zu beantworten, welche Faktoren in die Aufwandschätzung einbezogen werden müssen, um zuverlässige Schätzergebnisse zu gewährleisten.

Das dafür erforderliche Verständnis der Einflussfaktoren stellt sich, angesichts des Wesens der Integrationsvorhaben als komplexe, dynamische sozio-technische Systeme, als große Herausforderung dar. So ist von einer großen Anzahl unterschiedlicher Faktoren mit oftmals nicht-linearen und zeitverzögerten Wirkungen auf den Entwicklungsaufwand auszugehen.¹⁷ Im Zuge der notwendigen Komplexitätsreduktion müssen diejenigen Faktoren identifiziert werden, welche den Entwicklungsaufwand besonders signifikant beeinflussen.

1.1.3 Forschungsproblem

Fundierte empirische Erkenntnisse darüber, welche Faktoren den Entwicklungsaufwand der Anwendungsintegration determinieren, sind nicht bekannt. Diese sind jedoch erforderlich, um die aufgestellte Forschungsfrage beantworten zu können und damit zur Lösung des erläuterten praktischen Problems beizutragen.

Die unzureichende wissenschaftliche Durchdringung des Themenbereichs reicht diesbezüglich bis auf die unterste Ebene: Es fehlen bereits die Grundlagen für die Generierung geeigneter Hypothesen. Dieses Forschungsproblem wird im Weiteren erörtert.

Der Entwicklungswand und die auf ihn wirkenden Einflussfaktoren werden als reales Phänomen angesehen, dessen empirischer Nachweis im Allgemeinen über die Messung der Faktoren und dem analytischen Schluss auf die bestehenden Wirkungszusammen-

¹⁶ Vgl. Seibt /Projektaufwandschätzung/ 47.

¹⁷ Vgl. Milling /Komplexe Systeme/ 3. Diese Vermutung basiert auf den bisherigen Erkenntnissen zur Softwareentwicklung. Beispielsweise berichten Noth und Kretzschmar von einer Studie, die mit dem Ziel, zu stärker operativen und besser quantifizierbare Kriterien zur Bestimmung der Einflussfaktoren auf den Entwicklungsaufwand zu gelangen, letztlich in einem Katalog von 1200 Einflussfaktoren mündete (vgl. Noth, Kretzschmar /Aufwandschätzung/ 7.).

hänge erfolgt. Seit mehr als zwei Jahrzehnten werden wissenschaftliche Methoden zur Entwicklung valider¹⁸ und zuverlässiger¹⁹ Maße und zur empirischen Untersuchung von Wirkungszusammenhängen im Kontext der Softwareentwicklung diskutiert.²⁰ Dennoch wird der wissenschaftliche Standard der empirischen Studien in dieser Disziplin noch immer als gering eingeschätzt.²¹ Briand, Morasca und Basili fordern beispielsweise, dass ausgehend von klar spezifizierten Untersuchungszielen vier Schritte durchlaufen werden:²²

1. die Generierung empirischer Hypothesen,
2. die Definition von Maßen für die betrachteten Attribute,
3. die Konkretisierung der Hypothesen und
4. die Überprüfung der Hypothesen.

Empirische Hypothesen sind Aussagen bezüglich einer angenommenen Beziehung zwischen einem oder mehreren unabhängigen Attributen von Entitäten der Softwareentwicklungsvorhaben und einem abhängigen Attribut.²³ Sie repräsentieren das Verständnis eines Phänomens, dessen Gültigkeit überprüft werden soll.²⁴

Ausgehend von den empirischen Hypothesen gilt es, geeignete Maße²⁵ für die betrachteten unabhängigen Attribute und das abhängige Attribut zu bestimmen.²⁶ Dies erfordert die Formalisierung der betrachteten Attribute, die Identifikation geeigneter Abstraktio-

¹⁸ Ein Maß gilt als valide, wenn es genau das Attribut misst, welches es vorgibt zu messen (vgl. El Emam, Madhavji /Measuring the Success/ 203).

¹⁹ Ein Maß gilt als zuverlässig (engl. reliable), wenn der Anteil von zufälligen Messfehlern unter einem bestimmten gemeinsam akzeptierten Schwellenwert ist (vgl. El Emam, Madhavji /Measuring the Success/ 203).

²⁰ Vgl. beispielsweise Briand, Morasca, Basili /Measures/; Basili, Caldiera, Rombach /GQM/; Kitchenham, Pfleeger, Fenton /Measurement Validation/; Kitchenham u. a. /Empirical Research/; Basili, Weiss /Valid Software Engineering Data/.

²¹ Vgl. Kitchenham u. a. /Empirical Research/ 721.

²² Vgl. diesbezüglich Briand, Morasca, Basili /Measures/.

²³ Entitäten und Attribute stellen das Schema zur systematischen Analyse der realen Welt dar. Entitäten sind die in der realen Welt beobachtbaren Objekte, z. B. Produkte, Prozesse oder Ressourcen. Attribute sind die Eigenschaften, die eine Entität besitzt (vgl. Kitchenham, Pfleeger, Fenton /Measurement Validation/ 930).

²⁴ Vgl. Briand, Morasca, Basili /Measures/ 1114.

²⁵ Ein Maß definiert eine Vorschrift zur Bewertung der Ausprägungen eines Attributes (vgl. Kitchenham, Pfleeger, Fenton /Measurement Validation/ 930).

²⁶ Vgl. Briand, Morasca, Basili /Measures/ 1115.

nen der Attribute,²⁷ die Definition der Maße, die Definition des Messverfahrens und die theoretische und empirische Validierung der Maße.²⁸

Anhand der definierten Maße können die zugrunde gelegten empirischen Hypothesen im nächsten Schritt präzisiert werden. Insbesondere die angenommenen Beziehungen der Attribute können oftmals konkretisiert werden.²⁹

Im letzten der abgegrenzten Schritte kann nunmehr die empirische Überprüfung der Hypothesen erfolgen. Dazu muss ein angemessener Datenerhebungsprozess spezifiziert und durchgeführt werden.³⁰ Insbesondere gilt es, alle Faktoren zu identifizieren und zu erfassen, welche die Generalisierbarkeit der Ergebnisse und deren Interpretation betreffen können.³¹ Die erhobenen Daten werden anschließend anhand geeigneter Verfahren analysiert und die gewonnenen Ergebnisse interpretiert.³²

Angesichts dieses geforderten Vorgehens erscheint die Generierung angemessener empirischer Hypothesen grundlegend für die Entwicklung zweckmäßiger, valider und reliabler Maße. In ihnen kommt die Auswahlentscheidung zum Ausdruck, welche Entitäten und welche ihrer Attribute untersucht werden. Die Annahmen zu den Beziehungen der unabhängigen und abhängigen Attribute bilden den Unterbau für die Definition der Maße ebenso wie für die Erhebung und Analyse der Daten. Sie determinieren, welche empirischen Erkenntnisse überhaupt gewonnen werden können.

Es erscheint daher als geboten, dass die empirischen Hypothesen anhand eines umfassenden Verständnisses der Arbeitsvorgänge, Methoden und Technologien generiert werden, die in dem betrachteten Umfeld verwendet werden. Es wird empfohlen, auf dem Wissen von Experten der untersuchten Domäne oder auf fundierten Theorien aufzubauen. Schließlich wird gefordert, sowohl die formulierten Hypothesen als auch ihre Ursprünge explizit zu machen, damit sie diskutiert, hinterfragt und angepasst werden können.³³

²⁷ Eine Abstraktion ist eine mathematische Repräsentation einer Entität, so dass diese analysierbar und die relevanten Attribute quantifizierbar werden (vgl. Briand, Morasca, Basili /Measures/ 1116).

²⁸ Vgl. Briand, Morasca, Basili /Measures/ 115-1120; Kitchenham, Pfleeger, Fenton /Measurement Validation/ 930-940.

²⁹ Vgl. Briand, Morasca, Basili /Measures/ 1121.

³⁰ Vgl. Briand, Morasca, Basili /Measures/ 1121.

³¹ Vgl. Kitchenham u. a. /Empirical Research/ 723.

³² Vgl. Kitchenham u. a. /Empirical Research/ 729-732; Briand, Morasca, Basili /Measures/ 1122f.

³³ Vgl. Briand, Morasca, Basili /Measures/ 1114.

Insbesondere im Zusammenhang mit den bestehenden Methoden zur Schätzung des Entwicklungsaufwands für Softwaresysteme wird jedoch kritisiert, dass den durchgeführten empirischen Studien oftmals keine angemessenen Hypothesen zugrunde liegen.³⁴ Auf diese Weise aufgestellte prognostische Modelle gründen nicht auf einem theoretischen Verständnis der Einflussfaktoren. Die gemessenen Wirkungsbeziehungen zwischen den betrachteten Faktoren und dem Entwicklungsaufwand können daher nur schwerlich interpretiert werden. Die darauf hin aufgestellten Modelle sind lediglich anhand der erhobenen historischen Daten kalibriert und somit nur bedingt geeignet, zukünftige Situationen zutreffend zu beschreiben. Darüber hinaus werden sie oftmals als unvollständig angesehen. Anscheinend werden viele Faktoren mit signifikantem Einfluss auf den Entwicklungsaufwand in ihnen nicht berücksichtigt, was ebenfalls auf Mängel der zugrunde liegende Hypothesen zurückgeführt werden kann. Die anhand solcher Modelle gewonnenen Schätzergebnisse erweisen sich daher oftmals als unzuverlässig.

Auch im Falle der Anwendungsintegration besteht das Problem nicht darin, dass gegenwärtig überhaupt keine Hypothesen zu den Einflussfaktoren auf den Entwicklungsaufwand aufgestellt oder Maße vorgeschlagen werden könnten. Womöglich könnten in empirischen Studien auch einige signifikante Zusammenhänge festgestellt werden. Eine fundierte Interpretation dieser Ergebnisse wäre jedoch nicht möglich. Darüber hinaus bliebe im Unklaren, inwiefern davon ausgegangen werden kann, dass die zentralen Determinanten des Entwicklungsaufwands vollständig erfasst und verstanden wurden.

In diesem Sinne betonen Briand, Morasca und Basili: „The value added by the definition of a new measure is not the measure itself, but the fact that there is a theory in which the new measure is used to help explain some phenomenon of interest.“³⁵

Um empirische Erkenntnisse darüber zu erlangen, welche Faktoren den Entwicklungsaufwand der Anwendungsintegration determinieren, um valide und zuverlässige Maße für die Einflussfaktoren definieren zu können und ihre Wirkungsbeziehungen zu verstehen, gilt es demnach zunächst geeignete empirische Hypothesen zu generieren. Angesichts des Stands der wissenschaftlichen Erkenntnisse zur Anwendungsintegration er-

³⁴ Zu den im Weiteren angeführten Kritikpunkten vgl. Kitchenham, Pfleeger, Fenton /Measurement Validation/ 935-937.

³⁵ Briand, Morasca, Basili /Measures/ 1107.

scheint dies bisher jedoch als nicht möglich. Zwar werden in der Fachliteratur verschiedene Aspekte der Anwendungsintegration behandelt, die wissenschaftliche Auseinandersetzung mit diesem Themenbereich ist jedoch noch immer durch ein oftmals unklares und uneinheitliches Begriffsverständnis gekennzeichnet. Vor allem im Hinblick auf die Teilaufgaben der Anwendungsintegration erscheinen die Ausführungen als wenig differenziert oder gar unvollständig. Auch die Ableitung von Hypothesen aus den Erkenntnissen zu anderen Teildisziplinen der Softwareentwicklung unterliegt Restriktionen. Diese erscheinen nicht oder nur eingeschränkt auf die charakteristischen Integrationsvorhaben übertragbar. Den spezifischen Merkmalen der Anwendungsintegration dürften sie kaum gerecht werden, doch mit Sicherheit kann auch diese Frage bisher nicht beantwortet werden.

1.2 Erkenntnisziel der Arbeit

Vor dem Hintergrund der diskutierten praktischen Problemstellung und der wenig fortgeschrittenen wissenschaftlichen Auseinandersetzung mit der Thematik soll diese Arbeit einen ersten Beitrag zur systematischen Untersuchung der Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration leisten. Die Ansichten von erfahrenen Praktikern zu den Determinanten des Entwicklungsaufwands sollen erhoben werden, um damit die empfohlenen Grundlagen für die Generierung geeigneter Hypothesen zu schaffen.

Zu diesem Zweck können verschiedene Strategien verfolgt werden. Beispielsweise könnte auf der Grundlage der Fachliteratur eine Liste potenzieller Einflussgrößen erarbeitet werden, die anschließend schrittweise durch erfahrene Praktiker beurteilt, modifiziert und ergänzt wird. Diese Vorgehensweise wird beispielsweise von Dyba für die Untersuchung der Erfolgsfaktoren für die Verbesserung des Softwareentwicklungsprozesses (engl. software process improvement) beschrieben.³⁶ Zur Lösung des in dieser Arbeit aufgezeigten Forschungsproblems erscheint sie jedoch nicht als geeignet. Wie bereits erläutert wurde, liefert die Fachliteratur gegenwärtig keine angemessene Basis für die Generierung einer angemessenen Liste potenzieller Einflussfaktoren. In der Folge könnten die Auskünfte der Praktiker in ungewollter Weise eingeschränkt oder beeinflusst werden.

³⁶ Vgl. Dyba /Software Process Improvement/ 368f.

Alternativ könnten Dokumente zu durchgeführten Integrationsvorhaben analysiert werden, um zu erfahren, welche Einflussgrößen in welcher Weise von den Praktikern berücksichtigt werden. Diese Vorgehensweise wird beispielsweise von El Emam und Madhavji in die Konzeption eines Erfolgsmaßes für das Requirement Engineering einbezogen.³⁷ Auch dieser Ansatz erscheint zur Lösung des erläuterten Forschungsproblems nicht geeignet. Schließlich gründet dieses in der Erkenntnis, dass die Schätzung des Entwicklungsaufwands der Anwendungsintegration gegenwärtig nur unzureichend methodisch unterstützt wird. Es ist daher nicht davon auszugehen, dass in der Praxis die einer Aufwandschätzung zugrunde gelegten Annahmen und Informationen in differenzierter Weise dokumentiert werden. Diese Vermutung wird durch die Ergebnisse verschiedener Studien gestützt, wonach die überwiegende Mehrheit der Aufwandschätzungen in Softwareentwicklungsvorhaben auf „nicht expliziten, nicht wiederherstellbaren Schlussfolgerungen, d. h. auf Intuition“³⁸ basieren.³⁹

Es ist daher das Anliegen dieser Arbeit, diese Intuition erfahrener Praktiker zu ergründen. Ihr Verständnis von den maßgeblichen Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand soll in systematischer Weise erhoben werden. Dabei soll insbesondere ein Einblick in ihre Interpretationsmuster gewonnen werden, um zu verstehen, welche Schlussfolgerungen sie bei der Schätzung des Entwicklungsaufwands ziehen.

Diese Arbeit stellt somit lediglich einen ersten, wenngleich notwendigen Schritt zur Konzeption geeigneter Methoden zur Schätzung des Entwicklungsaufwands von Integrationsvorhaben dar. Die angestrebten Erkenntnisse sollen insbesondere eine Basis für die Generierung empirischer Hypothesen schaffen. Darüber hinaus wird angestrebt, bereits erste Anhaltspunkte für die praktische Aufwandschätzung von Integrationsvorhaben zu gewinnen.

1.3 Wissenschaftstheoretische Einordnung der Arbeit

Die Wirtschaftsinformatik versteht sich als anwendungsorientierte, interdisziplinäre Realwissenschaft, deren Bezugsrahmen durch die innerbetriebliche, die zwischenbetriebliche und die überbetriebliche Informationsverarbeitung in Organisationen gebildet

³⁷ Vgl. El Emam, Madhavji /Measuring the Success/ 206.

³⁸ Jorgensen /Expert Estimation/ 37.

³⁹ Vgl. Jorgensen /Expert Estimation/ 37-39.

wird.⁴⁰ Gegenstand des wissenschaftlichen Interesses der Wirtschaftsinformatik ist somit u. a. die Entwicklung von Informations- und Kommunikationssystemen, die als „Mensch-Computer- oder sozio-technische Systeme im jeweils individuellen organisatorischen Kontext“⁴¹ verstanden werden und denen die betrieblichen Anwendungssysteme als Komponenten zugeordnet werden können.⁴² Infolge dessen gehört die im Rahmen dieser Arbeit untersuchte Anwendungsintegration zum Gegenstandsbereich der Wirtschaftsinformatik.

Als anwendungsorientierte Wissenschaft verfolgt die Wirtschaftsinformatik als wichtiges oder gar als wichtigstes Ziel die Gewinnung von über Einzelfälle hinausgehenden Aussagen zur Lösung relevanter Entscheidungsprobleme in der Praxis. Empirische Forschung im Sinne der systematischen und nachvollziehbaren Erfassung und Analyse der Realität stellt somit einen unverzichtbaren Komplex zur Erreichung dieses Ziels dar.⁴³ Die Wirtschaftsinformatik kann daher zumindest in wesentlichen Teildisziplinen als empirische Wissenschaft aufgefasst werden.⁴⁴

Im Kontext der empirischen Forschung können verschiedene Strategien verfolgt werden. Diesbezüglich findet die Falsifikationsstrategie im Rahmen der Wirtschaftsinformatik häufige Verwendung. Basierend auf dem Ansatz des Kritischen Rationalismus strebt sie nach einer Sicherung bestehender Erkenntnis.⁴⁵ Als von größerer Bedeutung für den wissenschaftlichen Fortschritt kann demgegenüber jedoch die Explorationsstrategie angesehen werden, welche das Ziel eines wachsenden Verständnisses und der dadurch unter Umständen möglichen besseren Beherrschung der Realität verfolgt.⁴⁶ Die Exploration basiert auf der „informationalen Ausschöpfung von systematisch gewonnenem Erfahrungswissen zum Zweck der Theoriebildung.“⁴⁷ In diesem Sinne wird die empirische Forschung als iterativer Lernprozess verstanden, der auf einer „ständigen

⁴⁰ Vgl. Lehner /Wirtschaftsinformatik/ 505f.

⁴¹ Seibt /Anwendungssystem/ 47 (Die Hervorhebungen im Original wurden weggelassen.).

⁴² Vgl. Lehner /Wirtschaftsinformatik/ 506; Seibt /Anwendungssystem/ 47.

⁴³ Vgl. Kubicek /Heuristische Bezugsrahmen/ 5.

⁴⁴ Vgl. Holl /Wirtschaftsinformatik/ 168.

⁴⁵ Vgl. Müller-Böhlting /Organisationsforschung/ 1494. Die Falsifikationsstrategie wird auch als „Prüfstrategie empirischer Forschung“ bezeichnet (vgl. Kubicek /Heuristische Bezugsrahmen/ 6f.).

⁴⁶ Für die Explorationsstrategie wird in der Literatur auch der Begriff des „aufgeklärten konstruktiven Empirismus“ verwendet (vgl. Kubicek /Heuristische Bezugsrahmen/ 7f.).

⁴⁷ Wollnik /Erfahrungswissen/ 44.

Interaktion zwischen Fragen an die Realität, der Sammlung von Daten und der kritischen Reflexion des so gewonnen Realitätsbildes auf dem Hintergrund von aus dem Vorverständnis abgeleiteten Interpretationsmustern⁴⁸ basiert.

Aufgrund des bereits angesprochenen Mangels an empirischen Erkenntnissen zur Anwendungsintegration und des unzureichenden Verständnisses der Einflussfaktoren auf den Entwicklungsaufwand verfolgt die vorliegende Arbeit die Explorationsstrategie. Sie beruht auf einem qualitativen Forschungsansatz und versucht die subjektiven Sichtweisen und Erfahrungen erfahrener Praktiker zu verstehen.⁴⁹ Sie orientiert sich dabei an der Perspektive des symbolischen Interaktionismus, gemäß der die unterschiedlichen Weisen, in denen Subjekte Gegenstände, Ereignisse etc. mit Bedeutung versehen, zum zentralen Ansatzpunkt der Forschung werden.⁵⁰

1.4 Begriffliche Grundlagen

Zum Themenbereich der Integration betrieblicher Anwendungssysteme existiert gegenwärtig eine Vielzahl unterschiedlicher Begriffsverständnisse und Bezeichnungen.⁵¹ Die Ergebnisse einer empirischen Studie zeigen auf, dass es selbst auf dem Gebiet der Anwendungsintegration erfahrenen Personen schwer fällt, sie durch die Nennung von konstitutiven Merkmalen zu erklären. Sie beziehen sich überwiegend auf die angestrebten Ergebnisse bzw. den erhofften Nutzen.⁵²

Es ist daher von besonderer Bedeutung, dass das dieser Arbeit zugrunde liegende Begriffsverständnis und die verwendeten Bezeichnungen einführend definiert werden. Dazu wird zunächst der Begriff der betrieblichen Anwendungssysteme erläutert. Anschließend wird der Begriff der Integration eingeführt. Schließlich werden diese beiden Begriffe zusammengeführt und der Begriff der Anwendungsintegration definiert. In anschließenden Schritten wird die Anwendungsintegration gegenüber anderen Integrationsansätzen abgegrenzt und als Ausprägung der Softwareentwicklung beschrieben.

⁴⁸ Kubicek /Heuristische Bezugsrahmen/ 15.

⁴⁹ Zur Abgrenzung der qualitativen Forschung gegenüber der quantitativen Forschung siehe beispielsweise Flick /Qualitative Forschung/ 10-16.

⁵⁰ Vgl. Flick /Qualitative Forschung/ 29f.

⁵¹ Vgl. beispielsweise Themistocleous, Irani /Taxonomy for Application Integration/ 155f.; Heinrich, Heinzl, Roithmayr /Wirtschaftsinformatik-Lexikon/ 333; Lam /Enterprise Integration/ 292 sowie Lee, Siau, Hong /Enterprise Integration/ 56-59.

⁵² Vgl. Singletary /Application Integration/ 58f.

Als weitere zentrale Begriffe dieser Arbeit werden schließlich Entwicklungsaufwand und Aufwandschätzung definiert und abgegrenzt.

1.4.1 Herleitung einer Definition der Anwendungsintegration

Betriebliche Anwendungssysteme lassen sich als Softwaresysteme definieren, gebildet aus der Gesamtheit aller Programme, die als Anwendungssoftware einem konkreten betrieblichen Anwendungsgebiet dienen, und der zugehörigen Daten sowie der zu ihrem Einsatz erforderlichen Systemsoftware und systemnahen Software.⁵³ Betriebliche Anwendungssysteme sind ihrerseits Komponenten von betrieblichen Informationssystemen, im Sinne von „[...] Mensch-Computer- oder sozio-technische[n] Systeme[n] im jeweils individuellen organisatorischen Kontext.“⁵⁴

Die Bezeichnungen betriebliches Anwendungssystem und Anwendungssystem werden im Rahmen dieser Arbeit, wie auch im allgemeinen Sprachgebrauch der Wirtschaftsinformatik üblich, synonym verwendet.

Der Begriff der *Integration* spielt seit langem eine zentrale Rolle in der Wirtschaftsinformatik.⁵⁵ Er bedeutet im Allgemeinen, dass ein Ganzes durch das Zusammenführen logisch zusammenhörender Komponenten gebildet wird (das Wort ist abgeleitet vom lateinischen *integrare* = heil, unversehrt machen, wiederherstellen, ergänzen).⁵⁶

⁵³ Aufgrund der Relevanz der unterschiedlichen Charakteristika der Anwendungssysteme hinsichtlich der zum Betrieb der Anwendungssoftware erforderlichen Systemsoftware und systemnahen Software für die Anwendungsintegration, werden diese den Anwendungssystemen zugezählt. Die für den Einsatz der Anwendungssysteme erforderlichen Hardwarekomponenten werden nicht explizit in der Definition aufgegriffen. Ihre Charakteristika werden jedoch indirekt durch die Systemsoftware erfasst. Die Benutzer werden im Rahmen dieser Arbeit ebenfalls nicht in die Systemgrenzen eingeschlossen (vgl. Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 226f.). Systemsoftware orientiert sich an den Merkmalen bestimmter Hardwarekomponenten und dient dazu, diese zu steuern und nutzbar zu machen. Die Systemsoftware für ein bestimmtes Rechnersystem wird üblicher Weise unter der Bezeichnung Betriebssystem zusammengefasst. Systemnahe Software ist zwischen der Anwendungssoftware und der Systemsoftware einzuordnen. Sie stellt auf der Basis der Systemsoftware Funktionen bereit, welche von einem konkreten betrieblichen Anwendungsgebiet unabhängig sind und von der Anwendungssoftware genutzt werden (vgl. Seibt /Systemsoftware/ 858; Mellis /Projektmanagement/ 4; Gerhardt /Softwareindustrie/ 48-50).

⁵⁴ Vgl. Seibt /Anwendungssystem/ 47. Die Hervorhebung im Original wurde weggelassen.

⁵⁵ Heilmann verdeutlicht dies beispielsweise anhand eines „exemplarischen Streifzugs“ durch verschiedene Nachschlagewerke (vgl. Heilmann /Integration/ 47).

⁵⁶ Vgl. Wahrig /Deutsches Wörterbuch/ 683; Heinrich, Heinzl, Roithmayr /Wirtschaftsinformatik-Lexikon/ 333; Mertens /Integrierte Informationsverarbeitung/ 1.

Der Begriff der Integration wird sowohl zur Beschreibung eines Zustandes als auch zur Beschreibung eines Vorgangs verwendet.⁵⁷ Wird die Integration als Zustand verstanden, beschreibt sie, inwiefern die einzelnen Komponenten zusammengeführt, d. h. miteinander verbunden oder vereinigt, sind.⁵⁸ Wird die Integration hingegen als Vorgang verstanden, bezeichnet sie den Übergang der Komponenten von einem Integrationszustand in einen anderen.⁵⁹

Aufbauend auf den zuvor erläuterten Begriffsverständnissen wird unter der *Integration betrieblicher Anwendungssysteme* die Zusammenführung mehrerer Anwendungssysteme verstanden, welche logisch zusammengehörende Komponenten betrieblicher Informationssysteme darstellen.

Anwendungssysteme können auf verschiedene Weisen zusammengeführt werden. Somit können verschiedene Integrationsansätze unterschieden werden, von denen die Anwendungsintegration den zentralen Untersuchungsgegenstand dieser Arbeit darstellt.

Anwendungsintegration wird verstanden als die nachträgliche Verknüpfung von betrieblichen Anwendungssystemen eines oder mehrerer Unternehmen, um automatisierte Interaktionen der Anwendungssysteme zur systemübergreifenden Unterstützung von Geschäftsprozessen zu realisieren. Diese Interaktionen können darauf basieren, dass Anwendungssysteme die Funktionen anderer Anwendungssysteme nutzen oder Daten zwischen den Anwendungssystemen ausgetauscht werden. Ändernde oder erweiternde Eingriffe in die betroffenen Anwendungssysteme werden bei der Anwendungsintegration weitgehend vermieden. Insbesondere verbleiben die integrierten Anwendungssysteme selbständig lauffähig.⁶⁰

Dieser Arbeit liegt somit ein umfassendes Begriffsverständnis der Anwendungsintegration zugrunde, in das verschiedene in der Fachliteratur behandelte Konzepte, wie Enter-

⁵⁷ Zur weiteren Unterscheidung von Integrationszustand und Integrationsvorgang vgl. Linß /Nutzeffekte/ 5-7.

⁵⁸ In der deutschsprachigen Literatur existieren beispielsweise verschiedene Ansätze, anhand derer die Integration der Informationsverarbeitung im Sinne eines Zustandes systematisiert werden soll. Dazu werden jeweils verschiedene Merkmale zur Charakterisierung der Integrationsform unterschieden. Anhand der Ausprägungen dieser Merkmale soll dann im Einzelfall ein Integrationsgrad bestimmt werden können. Einen Überblick über verschiedene Ansätze zur Beschreibung von Integrationsformen bieten beispielsweise Linß (vgl. Linß /Nutzeffekte/ 7-17) sowie Mertens und Holzer (vgl. Mertens, Holzer /Integrationsansätze/ 5-24).

⁵⁹ Vgl. Kaib /EAI/ 10f.

⁶⁰ Vgl. Themistocleous, Irani /Taxonomy for Application Integration/ 156; Linthicum /EAI/ 3; Linthicum /Application Integration/ 1; Ruh, Maginnis, Brown /EAI/ 2; Wrazen /EAI/ 3.

prise Application Integration (EAI) oder Business-to-Business Integration, eingegliedert werden können.⁶¹

1.4.2 Abgrenzung der Anwendungsintegration gegenüber anderen Integrationsansätzen

Die Anwendungsintegration unterscheidet sich von einem Integrationsansatz, der als *portalorientierte Integration* bezeichnet wird.⁶² Dieser basiert darauf, die Benutzerschnittstellen mehrerer Anwendungssysteme zusammenzuführen. Dies wird i. d. R. aufgrund von Internettechnologien in Form eines sog. Portals erreicht. Die einzelnen Anwendungssysteme können bei der portalorientierten Integration durch einen Benutzer als einheitliches, d. h. ‚integriertes‘ System wahrgenommen werden. Eine automatisierte Interaktion der Anwendungssysteme im Sinne der Anwendungsintegration wird jedoch nicht ermöglicht.

Die Anwendungsintegration unterscheidet sich ferner von dem Ansatz, ein Integrationsproblem bereits während der Konzeption eines Anwendungssystems, d. h. zeitlich noch vor dessen Implementierung in einem Unternehmen, zu lösen. Dieser Ansatz wird im Weiteren als *Integration ex ante* bezeichnet. Dabei wird versucht, einen potenziellen Integrationsbedarf einmalig durch eine geeignete Gestaltung des Funktionsumfangs eines Anwendungssystems zu vermeiden. Alternativ kann ein Anwendungssystem in Form mehrerer integrationsfähiger Module konzipiert werden, die zusammengefügt werden können, um den Funktionsumfang schrittweise zu erweitern.⁶³ Als Beispiel können diesbezüglich die Enterprise Resource Planning (ERP)-Systeme⁶⁴ genannt werden, die meist als modularisierte Softwarepakete (engl. *pacaged systems*) verschiedene Anwendungsbereiche unterstützen.⁶⁵ Die Anwendungsintegration hingegen reprä-

⁶¹ Vgl. Themistocleous, Irani /Taxonomy for Application Integration/ 157-163.

⁶² Zur portalorientierten Integration vgl. im Weiteren Linthicum /Application Integration/ 19.

⁶³ Vgl. Kurbel, Rautenstrauch /Integration Engineering/ 169f.

⁶⁴ Da die ERP-Systeme insbesondere die Geschäftsprozesse eines Unternehmens und weniger die Ressourcen geschweige denn deren Planung betonen, erscheint die gebräuchliche Bezeichnung aus heutiger Sicht als ungeeignet (vgl. Mertens /ERP/ 183).

⁶⁵ Vgl. Lee, Siau, Hong /Enterprise Integration/ 56-60 oder Madapusi, D’Souza /ERP/ 7f. Gerade am Beispiel der ERP-Systeme wird deutlich, dass die Integration *ex ante* nur eingeschränkt zur Lösung des Integrationsproblems eines Unternehmens geeignet ist. Der Ersatz bestehender Anwendungssysteme durch ein ERP-System und die vollständige Einstellung deren Nutzung ist häufig nicht möglich oder wirtschaftlich sinnvoll. In diesen Fällen wird häufig der Ansatz der Anwendungsintegration gewählt um die ERP-Systeme mit den weiterhin bestehenden Anwendungssystemen zu verknüpfen (vgl. Kaib /EAI/ 21).

sentiert eine *Integration ex post*, bei der bestehende Anwendungssysteme nachträglich verknüpft werden.⁶⁶

Die Integration betrieblicher Anwendungssysteme kann schließlich dahingehend charakterisiert werden, in welchem Maße die zu integrierenden Anwendungssysteme verändert werden. Diesbezüglich können die invasive und die nicht-invasive Integration unterschieden werden (im angloamerikanischen Sprachgebrauch als „intrusive integration“ und „non-intrusive integration“ bzw. als „invasive integration“ und „non-invasive integration“ bezeichnet).⁶⁷ Invasive Integration basiert auf einer Veränderung oder Erweiterung der Anwendungssysteme. Nicht-invasive Integration bedeutet dagegen die Verknüpfung der Anwendungssysteme, ohne dass diese geändert oder erweitert werden müssen. In diesem Fall wird lediglich auf die vorhandenen externen Schnittstellen der Anwendungssysteme zurückgegriffen, selbst wenn dadurch die Gestaltungsfreiräume bei der Lösung eines Integrationsproblems eingeschränkt werden.⁶⁸

Im Zuge der Anwendungsintegration werden verändernde oder erweiternde Eingriffe in die Anwendungssysteme weitgehend vermieden. Sollten sie dennoch erforderlich sein, dienen sie lediglich dazu, externe Schnittstellen der Anwendungssysteme zu schaffen, wie sie für die Verknüpfung notwendig erscheinen.⁶⁹ Insbesondere ist zu betonen, dass die im Zuge der Anwendungsintegration zusammengeführten Anwendungssysteme sowohl vorher als auch nachher selbständig lauffähig sind. Anwendungsintegration unterscheidet sich damit von anderen Integrationsansätzen, die tief greifende Veränderungen der Anwendungssysteme herbeiführen, beispielsweise indem diese bildlich gesprochen miteinander verschmolzen werden und somit nicht mehr selbständig lauffähig sind.

Jeder der genannten Integrationsansätze zeichnet sich durch spezifische Stärken und Schwächen aus. Ihre jeweilige Eignung für die Integration betrieblicher Anwendungssysteme kann lediglich im Bezug auf konkrete Situationen beurteilt werden. Häufig werden in den Unternehmen daher verschiedene Integrationsansätze parallel zur Unterstützung der betrieblichen Informationsverarbeitung eingesetzt. Für die vorliegende Arbeit wird im Weiteren angenommen, dass die Anwendungsintegration zur Deckung

⁶⁶ Vgl. Kaib /EAI/ 22.

⁶⁷ Vgl. beispielsweise Kaib /EAI/ 81; Fincham /Application Integration/ 23.

⁶⁸ Vgl. Fincham /Application Integration/ 23.

⁶⁹ Vgl. Fincham /Application Integration/ 23.

des Integrationsbedarfs geeignet und technisch realisierbar ist. Szenarien, für die ein anderer Integrationsansatz bevorzugt werden sollte, werden dagegen von der Betrachtung ausgeschlossen.

1.4.3 Anwendungsintegration als Ausprägung der Softwareentwicklung

Über die Abgrenzung verschiedener Integrationsansätze hinaus wird die Anwendungsintegration in der Fachliteratur aus unterschiedlichen Perspektiven betrachtet. Beispielsweise wird sie als Strategie angesehen, als umfassendes Management der Anwendungslandschaften beschrieben oder als Sammelbezeichnung für bestimmte Technologien verwendet.⁷⁰ In dieser Arbeit wird die Anwendungsintegration dagegen als spezielle Ausprägung der Softwareentwicklung verstanden.

Softwareentwicklung kann im Allgemeinen als Aufgabe verstanden werden, eine zur Lösung eines Problems geeignete Software zu entwickeln und bereitzustellen. Entsprechend wird Anwendungsintegration im Einklang mit der eingeführten Begriffsdefinition als Aufgabe angesehen, für ein Integrationsproblem eine geeignete Integrationslösung zu entwickeln und bereitzustellen.

Eine *Integrationslösung* wird definiert als ein System, gebildet aus den verknüpften Anwendungssystemen und der Gesamtheit der Software, welche automatisierte Interaktionen der Anwendungssysteme unterstützt sowie den zugehörigen Daten. Jung verwendet diesbezüglich die Bezeichnung des integrierten verteilten Anwendungssystems (IVAS), definiert als „ein Anwendungssystem, das aus einem oder mehreren bereits miteinander integrierten, eventuell auf verschiedenen Netzknoten angesiedelten Anwendungssystemen besteht und dessen Architektur und Dokumentation darauf ausgerichtet sind, die Integration weiterer Anwendungssysteme zu ermöglichen.“⁷¹

Der strukturelle Aufbau einer Integrationslösung wird als Integrationsarchitektur bezeichnet. In Anlehnung an den Begriff der Softwarearchitektur gemäß Bass, Clements und Kazman kann eine Integrationsarchitektur beschrieben werden anhand der Soft-

⁷⁰ Zur Charakterisierung der Anwendungsintegration als Strategie vgl. beispielsweise Linthicum /Application Integration/ 1. McKeen und Smith sprechen dagegen von der Anwendungsintegration als Komplex aus „Plänen, Methoden und Werkzeugen, welche darauf zielen, die Computeranwendungen innerhalb eines Unternehmens zu modernisieren, zu konsolidieren, zu integrieren und zu koordinieren“ (McKeen, Smith /Enterprise Application Integration/ 451). Zum Verständnis der Anwendungsintegration als Sammelbezeichnung für bestimmte Technologien vgl. Themistocleous, Irani /Taxonomy for Application Integration/ 154; Edwards, Newing /Application Integration/ 1.

⁷¹ Jung /Integrationsorientiertes Reengineering/ 5.

warekomponenten, aus denen die Integrationslösung besteht, der nach Außen sichtbaren Eigenschaften dieser Komponenten und der Beziehungen zwischen ihnen.⁷² Jung spricht in diesem Zusammenhang von einer Verteilungsarchitektur.⁷³

Die praktische Durchführung der Softwareentwicklung wird im Weiteren als Softwareentwicklungsvorhaben oder kurz als Entwicklungsvorhaben bezeichnet. Da diese im Allgemeinen durch die Einmaligkeit der herrschenden Bedingungen in ihrer Gesamtheit gekennzeichnet sind (z. B. im Hinblick auf die Zielvorgabe oder zeitliche, finanzielle und personelle Begrenzungen) werden sie auch als Projekt bezeichnet.⁷⁴ Ein Softwareentwicklungsvorhaben, das den Ansatz der Anwendungsintegration verfolgt, wird im Weiteren als *Integrationsvorhaben* bezeichnet.

Es können verschiedene Arten von Entwicklungsvorhaben unterschieden werden. Hinsichtlich der zugrunde liegenden Aufgabenstellung wird zwischen einer Auftragsentwicklung und einer Marktentwicklung differenziert. Existiert ein Kunde (innerhalb oder außerhalb des Unternehmens, welches für die Softwareentwicklung verantwortlich ist), der die Entwicklung der Software gemäß seinen Bedürfnissen und Anforderungen beauftragt hat, so handelt es sich um eine Auftragsentwicklung. Eine Marktentwicklung liegt dagegen vor, wenn das herstellende Unternehmen die Software auf eigenes Risiko entwickelt, um sie später meist stark standardisiert am Markt anzubieten.⁷⁵

Da die Anwendungsintegration die Verknüpfung konkreter Anwendungssysteme eines oder mehrerer Unternehmen bezeichnet, wird im Weiteren stets von Auftragsentwicklungen ausgegangen.

1.4.4 Entwicklungsaufwand und Aufwandschätzung

Das dieser Arbeit zugrunde liegende und im Kontext der Softwareentwicklung im Allgemeinen gebräuchliche Begriffsverständnis des *Aufwands* unterscheidet sich vom Verständnis des Aufwands aus betriebswirtschaftlicher Sicht.

⁷² “[The] software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.” (Vgl. Bass, Clements, Kazman /Software Architecture/ 21)

⁷³ Vgl. Jung /Integrationsorientiertes Reengineering/ 5.

⁷⁴ Vgl. DIN /DIN 69901/ 1.

⁷⁵ Vgl. Mellis /Projektmanagement/ 6.

In der Betriebswirtschaftslehre wird unter Aufwand der wertmäßige Verbrauch von Wirtschaftsgütern (Werteverzehr) einer Periode verstanden, der mit Ausgaben zusammenhängt. Dem Aufwand wird der Ertrag gegenüber gestellt, der den Wertzugang einer Periode bezeichnet, der zu Einnahmen führt.⁷⁶ Aufwendungen und Erträge drücken entsprechend Wertebewegungen aus. Neben Ausgaben und Einnahmen sowie Kosten und Leistungen werden sie im Rechnungswesen als Stromgrößen behandelt.⁷⁷

Im Unterschied dazu wird Aufwand in dieser Arbeit zum einen nicht als Wert-, sondern vielmehr als Mengengröße verstanden. Zum anderen wird der Begriff im Allgemeinen nicht auf jegliche Form von Wirtschaftsgütern, sondern in der Regel ausschließlich auf personelle Arbeitsleistung bezogen.⁷⁸ In diesem Sinne bezeichnet der mit einem Entwicklungsvorhaben verbundene Aufwand den mengenmäßigen Verbrauch von personeller Leistung für die Entwicklung und Bereitstellung eines Softwaresystems. Gemessen wird er gemeinhin als die Summe der Arbeitszeit, die Personen für die Erfüllung der mit dem Entwicklungsvorhaben verbundenen Aufgaben einsetzen. Ferner wird er zu meist in den Einheiten Personen-Tag, Personen-Monat oder Personen-Jahr ausgedrückt.⁷⁹ Es handelt sich dabei um künstliche Einheiten, die von einer einheitlichen und konstanten Arbeitsleistung einer Person pro Tag, pro Monat oder pro Jahr ausgehen.⁸⁰ Wird der so ausgedrückte Aufwand, der für die Entwicklung eines Softwaresystems angenommen wird, ins Verhältnis zur verfügbaren Entwicklungszeit gestellt, so zeigt dies theoretisch die Zahl der für die Entwicklung einzusetzenden Mitarbeiter.⁸¹ Diese gebräuchliche Schlussfolgerung und die zum Ausdruck des Entwicklungsaufwands verwendeten Einheiten erweisen sich jedoch als problematisch, da sie eine beliebige Teilbarkeit, Konstanz und Homogenität der menschlichen Leistung annehmen. Empirische Untersuchungen haben dagegen gezeigt, dass große Unterschiede und Schwankun-

⁷⁶ Vgl. Seibt /Projektaufwandschätzung/ 29; Schierenbeck /Betriebswirtschaftslehre/ 516.

⁷⁷ Vgl. Schierenbeck /Betriebswirtschaftslehre/ 515.

⁷⁸ Andere Aufwandsarten, wie beispielsweise Technik-/Rechner-/System-Kapazitäten, erscheinen gegenüber der herausragenden Bedeutung des Personalaufwands als Kostenfaktor für die Softwareentwicklung vernachlässigbar (vgl. Noth, Kretzschmar /Aufwandschätzung/ 4).

⁷⁹ Auch als Mitarbeiter- oder Manntag, -monat oder -jahr bezeichnet (vgl. Balzert /Software-Technik/ 74; Noth, Kretzschmar /Aufwandschätzung/ 4).

⁸⁰ Dabei werden potenzielle Urlaubs- und sonstige Fehlzeiten pauschal berücksichtigt, beispielsweise in dem ein Personenjahr oftmals neun oder zehn Personenmonaten gleichgesetzt wird (vgl. Balzert /Software-Technik/ 74). Boehm nennt konkretere Formeln zum Umrechnen zwischen den verschiedenen Einheiten. Ein Personenmonat umfasst demnach 19 Personentage oder 152 Personenstunden (vgl. Boehm /Software Engineering Economics/ 59). Siehe diesbezüglich auch Stevenson /Software Engineering Productivity/ 23-25.

⁸¹ Vgl. Balzert /Software-Technik/ 75.

gen bezüglich der Leistung einzelner Mitarbeiter vorliegen, die unter anderem auf die individuellen Fähigkeiten der Mitarbeiter zurückgeführt werden.⁸²

In dieser Arbeit wird lediglich die personelle Arbeitsleistung im Zusammenhang mit der Entwicklung und Bereitstellung eines Softwaresystems bzw. einer Integrationslösung betrachtet. Personelle Arbeitsleistungen, die mit dem Betrieb eines Softwaresystems bzw. einer Integrationslösung, d. h. der Administration, Wartung und Nutzung zusammenhängen, werden dagegen nicht betrachtet.⁸³ Um diese Abgrenzung zu betonen, wird im Weiteren insbesondere der Terminus *Entwicklungsaufwand* als Bezeichnung für den erläuterten Aufwandsbegriff verwendet.

Die *Aufwandschätzung* betrifft als Teilaufgabe des Projektmanagements die Vorhersage des mit größtmöglicher Wahrscheinlichkeit erforderlichen Entwicklungsaufwands. Sie kann von der Aufwandsplanung abgegrenzt werden. Während erstere den wahrscheinlich erforderlichen Aufwand für die Durchführung bestimmter Tätigkeiten bzw. die Fertigstellung bestimmter Arbeitsergebnisse voraussagen sucht, trifft letztere Vorgaben für den zulässigen Aufwand im Sinne eines Ziels und begründet damit i. d. R. Aufwandsbeschränkungen.⁸⁴

Die Aufwandschätzung sollte nicht als einmalig zu bearbeitende Aufgabe angesehen werden. Es wird vielmehr empfohlen, die Aufwandschätzung mehrfach zu verschiedenen Zeitpunkten im Projektverlauf durchzuführen. Durch das zunehmende Wissen über das Entwicklungsvorhaben ist dabei von einer zunehmenden Schätzgenauigkeit auszugehen. Die wiederholte Aufwandschätzung wird als wesentliche Grundlage für eine kontinuierliche Planung, Überwachung und Steuerung des Entwicklungsaufwands angesehen.⁸⁵

Wird der mit dem Entwicklungsaufwand verbundene Ressourcenverbrauch monetär bewertet, so erhält man die *personellen Entwicklungskosten*. Erst diese stellen somit eine Wertgröße dar. Dabei können die aus einem bestimmten Entwicklungsaufwand

⁸² Vgl. Brooks /Mythical Man Month/; Balzert /Software-Technik/ 75.

⁸³ Zu den Abschnitten des Lebenszyklus eines Softwaresystems vgl. Seibt /Systemlebenszyklus/ 456-458.

⁸⁴ Vgl. Seibt /Projektaufwandschätzung/ 48.

⁸⁵ Vgl. Jones /Software Costs/ 17-19.

resultierenden Kosten stark variieren, beispielsweise in Abhängigkeit von den Personalkostensätzen oder Währungswechselkursen bei internationalen Projekten.⁸⁶ Trotz dieses Unterschieds wird in der Fachliteratur zuweilen nicht von Aufwandschätzung, sondern von Kostenschätzung gesprochen, obwohl der nicht bewertete Aufwand im erläuterten Sinne thematisiert wird.⁸⁷

1.5 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich im Wesentlichen in fünf Schritte. Nachdem in diesem Kapitel die Problemstellung und das Erkenntnisziel der Arbeit definiert wurden, erfolgt in Kapitel 2 eine umfassende Beschreibung des konzeptionellen Bezugsrahmens der Untersuchung. Hierbei werden zunächst die grundlegenden Charakteristika von Integrationsproblemen, Integrationsarchitekturen und darauf aufbauend von Integrationsvorhaben vorgestellt.

In Kapitel 3 wird ein Einblick in den Stand der Erkenntnis zu den Einflussfaktoren auf den Entwicklungsaufwand gegeben. Es werden mehrere Untersuchungen und Verfahren zur Aufwandschätzung aus verschiedenen Teildisziplinen der Softwareentwicklung vorgestellt und diese vor dem Hintergrund des konzeptionellen Bezugsrahmens beurteilt.

In Kapitel 4 wird die Konzeption der daraufhin durchgeführten Untersuchung erläutert. Ausgehend von der Definition der Untersuchungsziele werden die theoretischen Grundlagen der Untersuchung dargestellt, die angewandte Untersuchungsmethode spezifiziert und der Ablauf der Untersuchung beschrieben.

In Kapitel 5 erfolgt die Darstellung der gewonnenen Untersuchungsergebnisse. Hierbei werden zunächst die hinter den Ergebnissen stehenden Untersuchungssubjekte und -objekte charakterisiert. Anschließend wird die angewandte Analysemethode erläutert, bevor die eigentlichen Untersuchungsergebnisse systematisch dargestellt und den bisherigen Erkenntnissen gegenübergestellt werden.

In Kapitel 6 schließlich erfolgt die Beurteilung der Untersuchung und der gewonnenen Ergebnisse vor dem Hintergrund der zugrunde liegenden Problemstellung und dem definierten Erkenntnisziel der Arbeit.

⁸⁶ Vgl. Boehm /Software Engineering Economics/ 61; Noth, Kretzschmar /Aufwandschätzung/ 4; Jones /Software Costs/ 13f.

⁸⁷ Siehe z. B. Balzert /Software-Technik/ 74.

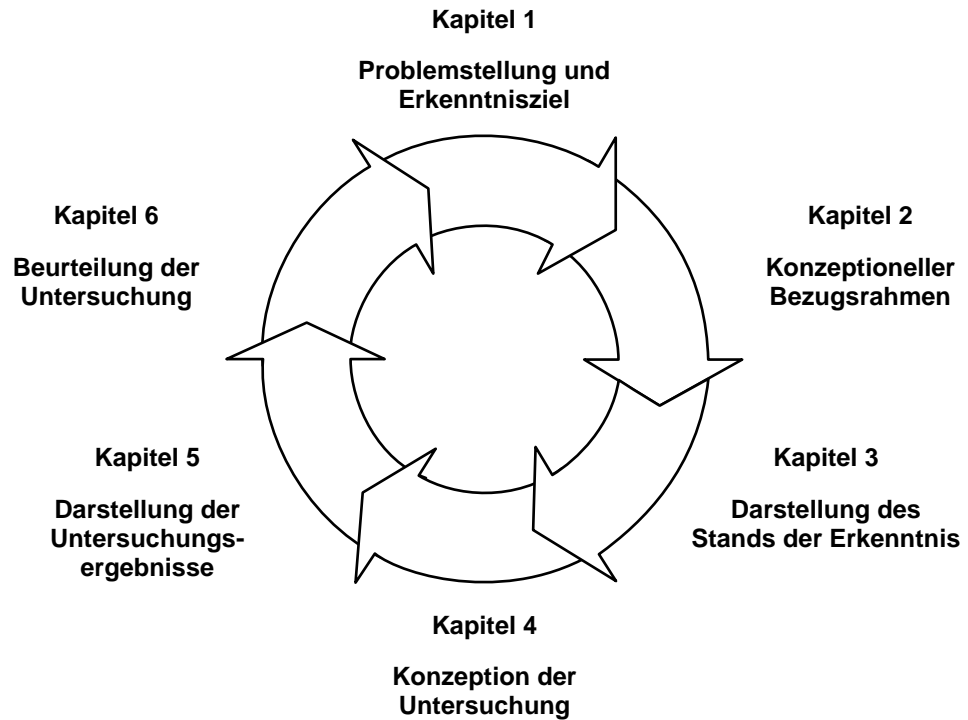


Abbildung 1-1: Aufbau der Arbeit

2 Konzeptioneller Bezugsrahmen der Untersuchung

Softwareentwicklung kann als Aufgabe verstanden werden, eine zur Lösung eines Problems geeignete Software zu entwickeln und bereitzustellen. Ausgehend von diesem Verständnis können drei Teilaspekte unterschieden werden, anhand derer die Anwendungsintegration als spezielle Ausprägung der Softwareentwicklung charakterisiert werden kann: das zugrunde liegende Problem, die realisierte Problemlösung sowie das Vorgehen, um von dem Problem zur Problemlösung zu gelangen.

Den Ausgangspunkt bildet ein Integrationszustand bestimmter Anwendungssysteme, der als unbefriedigend angesehen wird. Er wird im Weiteren als *Integrationsproblem* bezeichnet und erfordert eine *Integrationslösung*. Diese repräsentiert einen gegenüber dem Ausgangspunkt veränderten Integrationszustand der Anwendungssysteme. Die Entwicklung und Bereitstellung einer Integrationslösung erfolgt in Form eines Softwareentwicklungsvorhabens, das im Weiteren als *Integrationsvorhaben* bezeichnet wird. Ein Integrationsvorhaben kann in verschiedene Teilaufgaben untergliedert werden, deren Erfüllung mit personeller Arbeitsleistung und damit mit *Entwicklungsaufwand* verbunden ist.

Die Integration betrieblicher Anwendungssysteme stellt sich in der Praxis häufig nicht als ein einmaliges Vorhaben, sondern vielmehr als fortlaufender, d. h. zyklischer, Prozess dar. Dies ist insbesondere darauf zurückzuführen, dass sich die Unternehmen und ihre wirtschaftliche, technische und sozio-politische Umwelt im Zeitablauf verändern. Ein vormals als angemessen angesehener Integrationszustand kann somit zu einem späteren Zeitpunkt als Integrationsproblem wahrgenommen werden und damit den Anlass für ein weiteres Integrationsvorhaben bilden. Innerhalb dieses zyklischen Prozesses erfolgt die Anwendungsintegration nicht isoliert von anderen Integrationsbemühungen. Im Rahmen der betrachteten Integrationsvorhaben kann es somit erforderlich sein, frühere Maßnahmen zur Integration ihrerseits zusammenzuführen, die auf verschiedenen der vorgestellten Integrationsansätze basieren können.⁸⁸ Im Interesse einer klaren Abgrenzung der zu erläuternden Konzepte wird von diesem zyklischen Charakter bei der Erläuterung des konzeptionellen Bezugsrahmens weitgehend abstrahiert.

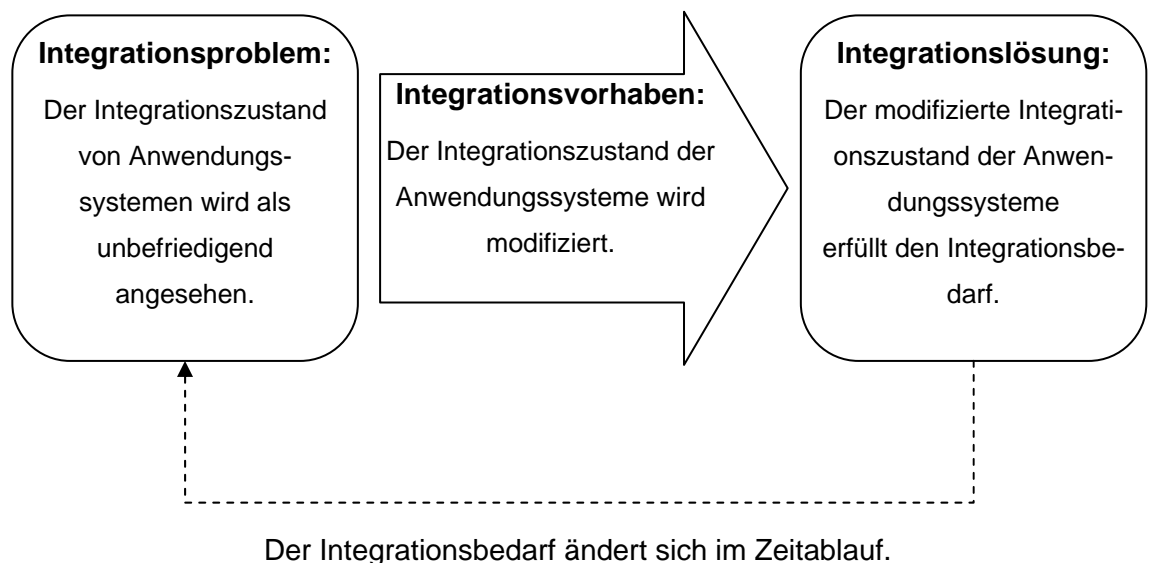


Abbildung 2-1: Zyklus von Integrationsproblem, Integrationsvorhaben und Integrationslösung

⁸⁸ Vgl. Altman /Integration Tools/ 36.

2.1 Integrationsprobleme

Unter einem Integrationsproblem wird in dieser Arbeit ein Integrationszustand von Anwendungssystemen verstanden, der als unzureichend angesehen wird, einen spezifischen Integrationsbedarf auf geeignete Weise zu erfüllen.

Bezogen auf eine automatisierte Interaktion zwischen Anwendungssystemen beschreibt dieses Problemverständnis eine mangelnde Kongruenz von Integrationsbedarf und Integrationszustand. Es betrifft damit einen Ausschnitt aus dem umfassenden Problem der Entwicklung von Informationssystemen, wie es aus Sicht der Informationsverarbeitungstheorie (engl. information processing theory) beschrieben wird. Diese betrachtet die Effektivität und Effizienz der betrieblichen Informationsverarbeitung als Folge des Ausmaßes an Übereinstimmung zwischen den spezifischen Bedürfnissen und Fähigkeiten der Informationsverarbeitung.⁸⁹

2.1.1 Anwendungssysteme als Abbildungen von Ausschnitten der betrieblichen Realität

Unternehmen können als marktorientierte Systeme verstanden werden, in denen zielgerichtete Entscheidungen gefällt, realisiert und kontrolliert werden. Dazu müssen Informationen gewonnen, verarbeitet und vermittelt werden (im Weiteren kurz als Informationsverarbeitung bezeichnet). Diese erforderliche Informationsverarbeitung ist in der Regel so komplex, dass sie arbeitsteilig erbracht und informations- und kommunikationstechnisch (im Weiteren kurz als informationstechnisch bezeichnet) unterstützt wird.⁹⁰

Ausgehend von diesem Verständnis können drei Gestaltungsebenen von Unternehmen unterschieden werden, welche die betriebliche Informationsverarbeitung prägen: die Ebene der Strategie, die Ebene der Geschäftsprozesse und die Ebene der Anwendungssysteme.⁹¹

⁸⁹ Vgl. beispielsweise Premkumar, Ramamurthy, Saunders /Information Processing/ 260-262 sowie die dort angeführten Quellen.

⁹⁰ Vgl. Frese /Organisation/ 5f.

⁹¹ Eine solche Strukturierung wird beispielsweise in dem Modell der ganzheitlichen Informationssystem-Architektur nach Krcmar vorgenommen, welches die Gestaltungsbereiche eine Unternehmensarchitektur und deren Zusammenhänge beschreibt (vgl. Krcmar /Informationsmanagement/ 41-43). Österle unterscheidet ebenfalls die genannten Ebenen zur Erläuterung des Konzeptes der integrierten Informationsverarbeitung (vgl. Österle /Integration/ 6-11).

Auf der Ebene der Strategie werden die langfristig gültigen Grundsatzentscheidungen eines Unternehmens getroffen, durch welche zukünftige Erfolgspotenziale aufgebaut und gesichert werden sollen. Wesentliche Aspekte einer Strategie bilden neben der Abgrenzung des für das Unternehmen relevanten Marktes die Festlegung des Verhaltens gegenüber den übrigen Marktteilnehmern sowie die Gestaltung der unternehmenseigenen Ressourcenpotentiale.⁹² Indem Geschäftsfelder ausgewählt, zwischenbetriebliche Kooperationen beschlossen und die Gestaltung des Produktionsfaktors Information langfristig ausgerichtet werden, legt die Geschäftsstrategie die grundlegenden Ziele der Informationsverarbeitung fest.⁹³

Diese strategischen Vorgaben werden in Abläufe der unternehmerischen Leistungserbringung (sog. Ablauforganisation) und organisatorische Strukturen (sog. Aufbauorganisation) umgesetzt.⁹⁴ Der Orientierung an Geschäftsprozessen wird bei dieser organisatorischen Gestaltung zunehmend eine besondere Bedeutung beigemessen.⁹⁵

Ein Geschäftsprozess kann als eine Folge von logisch zusammengehörigen Aktivitäten definiert werden, die in einer geeigneten Ablauflogik unter Verwendung bestimmter Informationen und ggf. weiterer Inputs von organisatorischen Einheiten so ausgeführt werden, dass eine angestrebte Leistung erbracht wird.⁹⁶ Diese Leistung kann materiell oder nicht-materiell sein und unternehmensintern oder unternehmensextern nachgefragt werden.⁹⁷ Ein Geschäftsprozess besitzt einen definierten Beginn, der durch ein Starterereignis ausgelöst wird, und ein definiertes Ende, welches durch ein Ergebnisereignis, d. h. die erbrachte Leistung repräsentiert wird.⁹⁸ Da auch die jeweiligen Ergebnisse der einzelnen Aktivitäten eines Geschäftsprozesses Leistungen darstellen, können die logisch zusammenhängenden Aktivitäten der Geschäftsprozesse mit unterschiedlichem Detaillierungsgrad abgegrenzt werden. Ein mehrstufiger Geschäftsprozess kann somit in mehrere Teilprozesse unterteilt werden.

⁹² Vgl. Frese /Organisation/ 284.

⁹³ Vgl. Frese /Organisation/ 128ff.; Österle /Integration/ 7.

⁹⁴ Vgl. Frese /Organisation/ 7.

⁹⁵ Vgl. Scheer /ARIS/ 3; Kirchmer /Standardsoftware/ 8.

⁹⁶ Vgl. Kirchmer /Standardsoftware/ 8; Scheer /ARIS/ 3.

⁹⁷ Vgl. Scheer /ARIS/ 13.

⁹⁸ Vgl. Kirchmer /Standardsoftware/ 8.

Die verschiedenen Geschäftsprozesse eines Unternehmens können in unterschiedlicher Weise zusammenhängen. Beispielsweise kann das Ergebnisereignis eines Geschäftsprozesses das Startereignis eines anderen Geschäftsprozesses bilden, die Leistung eines Geschäftsprozesses kann im Zuge der Abwicklung eines anderen Geschäftsprozesses nachgefragt werden oder ein Startereignis kann den Beginn mehrerer Geschäftsprozesse auslösen. Darüber hinaus können die Aktivitäten verschiedener Geschäftsprozesse von denselben organisatorischen Einheiten ausgeführt werden oder dieselben Informationen betreffen. Zusammenhänge zwischen Geschäftsprozessen können schließlich nicht nur innerhalb eines Unternehmens, sondern auch über mehrere Unternehmen hinweg bestehen.⁹⁹ Die interne sowie übergreifende Harmonisierung der Geschäftsprozesse in zeitlicher und räumlicher Hinsicht ist insbesondere Gegenstand der Ansätze des Business Process Reengineering (BPR)¹⁰⁰ und des Supply Chain Management (SCM).¹⁰¹

Auf der untersten der unterschiedenen Ebenen schließlich wird das Ziel verfolgt, die in den Geschäftsprozessen anfallende Informationsverarbeitung informationstechnisch zu unterstützen. Dazu werden Teilbereiche der Informationsverarbeitung als Anwendungsgebiete abgegrenzt und Anwendungssystemen zugeordnet.

Die Anwendungssysteme bilden jeweils Ausschnitte aus den Geschäftsprozessen und den zwischen ihnen bestehenden Zusammenhängen ab. Dabei repräsentieren sie ein spezifisches Verständnis der betrieblichen Informationsverarbeitung, u. a. im Hinblick auf die benötigten Informationen, die Aktivitäten und die Ablauflogik, was entspre-

⁹⁹ Vgl. Scheer /ARIS/ 10-53.

¹⁰⁰ Zum Business Process Reengineering siehe beispielsweise Davenport, Short /Business Process Redesign/.

¹⁰¹ Zum Supply Chain Management siehe beispielsweise Stadtler /Supply Chain Management/.

chend in Datenmodellen¹⁰², Funktionsmodellen¹⁰³ und Kontrollflussmodellen¹⁰⁴ zum Ausdruck kommt.¹⁰⁵

In den Unternehmen wird in der Regel eine Vielzahl verschiedenartiger Anwendungssysteme eingesetzt, die jeweils bestimmte Bereiche der betrieblichen Informationsverarbeitung unterstützen. Diese fragmentarischen und technisch heterogenen Anwendungslandschaften können auf verschiedene Ursachen zurückgeführt werden, wie z. B. die vorwiegend funktionsbereichsorientierte Konzeption der Anwendungssysteme,¹⁰⁶ die gezielte Kombination verschiedener Anwendungssysteme¹⁰⁷ oder ergänzende anstelle ersetzender Investitionen in neue Anwendungssysteme¹⁰⁸ vor dem Hintergrund immer kürzer werdender Innovationszyklen der Informations- und Kommunikationstechnik¹⁰⁹ sowie sich ändernder Entwicklungsparadigmen.¹¹⁰ Oftmals erscheint die Vielfalt der Anwendungssysteme in den Unternehmen schlicht als Folge einer unzureichenden

¹⁰² Datenmodelle beschreiben ein spezifisches Datenschema durch die Abgrenzung und Spezifikation von Datenelementen sowie der Definition der Beziehungen zwischen den Datenelementen (vgl. Elmasri, Navathe /Database Systems/ 27f.).

¹⁰³ Funktionenmodelle repräsentieren insbesondere das Verständnis der Aktivitäten. Sie beschreiben die einzelnen in dem Anwendungssystem implementierten Funktionen sowie ihre Beziehungen zueinander (vgl. Balzert /Software-Technik/ 116).

¹⁰⁴ Kontrollflussmodelle beschreiben die Steuerung des Ablaufs von Funktionen durch Ereignisse (vgl. beispielsweise Balzert /Software-Technik/ 208). Das Verständnis der Ablauflogik kommt zudem in der angewandten Nutzungsform, d. h. einer interaktiven Abwicklung von Datenverarbeitungsaufgaben oder einer Stapelverarbeitung (englisch: batch processing) sowie eingeplanten Stillstandszeiten der Anwendungssysteme zum Ausdruck (vgl. Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 79, 81; Britton /IT-Architecture/ 120).

¹⁰⁵ Die beschriebenen Aspekte der Geschäftsprozesse sowie ihre Modellierung und Umsetzung in Anwendungssystemen sind der zentrale Gegenstand des ARIS-Geschäftsprozessmodells nach Scheer (vgl. Scheer /ARIS/ 10-53).

¹⁰⁶ Vgl. Kirchmer /Standardsoftware/ 18f.

¹⁰⁷ Die einzelnen Anwendungssysteme, insbesondere Standardsoftwareprodukte, sind hinsichtlich der von ihnen bereitgestellten Funktionalität unflexibel und implizieren häufig konkrete organisatorische Gestaltungen. Die Unternehmen sind daher oftmals darauf angewiesen, sie durch eigenentwickelte Softwarelösungen zu ergänzen oder geeignete Teile verschiedener Standardsoftwareprodukte zu kombinieren (sog. „Best-of-breed“-Ansatz), um die benötigte Funktionalität bestmöglich abzudecken und strategische Differenzierungspotenziale zu bewahren (vgl. Davenport /Enterprise System/ 121f.; Kaib /EAI/ 56).

¹⁰⁸ Ältere Anwendungssysteme werden in diesen Fällen neben den neuen Anwendungssystemen weiter verwendet, wenngleich womöglich beschränkt auf einige Anwendungsfälle oder bestimmte Benutzergruppen. Als Begründung kann beispielsweise der Investitionsschutz genannt werden (vgl. Sandoe, Corbitt, Boykin /Enterprise Integration/ 27). Weitere mögliche Gründe für ergänzende anstelle ersetzender Investitionen sind die Vermeidung von zusätzlichem Migrationsaufwand, die Präferenzen einzelner Nutzergruppen oder gesetzliche Vorgaben (z. B. Aufbewahrungspflicht nach Handels- und Steuerrecht) (vgl. Kaib /EAI/ 13).

¹⁰⁹ Die Informations- und Kommunikationstechnik wird im Weiteren kurz Informationstechnik genannt.

¹¹⁰ Vgl. Sandoe, Corbitt, Boykin /Enterprise Integration/ 27; Österle /Integration/ 4; Kaib /EAI/ 13; Cummins /Enterprise Integration/ 2f.

Planung der Gesamtarchitektur der Anwendungslandschaften.¹¹¹ Es ist nicht abzusehen, dass die Vielfalt der im Rahmen der Informationsverarbeitung zu bewerkstellenden Aufgaben in den Unternehmen in naher Zukunft durch homogene Lösungen abgedeckt würden.¹¹²

2.1.2 Abhängigkeiten zwischen den Anwendungssystemen

Wie erläutert bilden die einzelnen Anwendungssysteme jeweils einen spezifischen Ausschnitt der Geschäftsprozesse und der zwischen ihnen bestehenden Zusammenhänge ab. Dabei kommt i. d. R. die implizite oder explizite Annahme zum Ausdruck, dass die einzelnen Ausschnitte unabhängig von einander seien. Dies ist jedoch oftmals nicht der Fall, so dass vielfältige Abhängigkeiten zwischen den Anwendungssystemen bestehen können.¹¹³

Diese Abhängigkeiten können zum einen aus Überschneidungen der abgebildeten Ausschnitte der Informationsverarbeitung resultieren. So können verschiedene Anwendungssysteme die gleichen Aktivitäten, die gleichen Informationen, die gleichen organisatorischen Einheiten oder die gleichen angestrebten Leistungen betreffen. Die Abhängigkeiten können zum anderen aus Zusammenhängen zwischen den abgebildeten Ausschnitten der Informationsverarbeitung resultieren. So wurde bereits erläutert, dass verschiedene Geschäftsprozesse über die Start- und Ergebnisereignisse sowie die Leistungen in Beziehung zueinander stehen können. Wenn diese Geschäftsprozesse durch verschiedene Anwendungssysteme unterstützt werden, können diese Beziehungen zu Abhängigkeiten im Einsatz der Anwendungssysteme führen.

¹¹¹ So wurde und wird in vielen Unternehmen lediglich auf der Grundlage von informellen Plänen, un-abgestimmten Einzelprojektplänen oder in kurzfristiger Reaktion auf vorgegebene Unternehmenspläne über den Einsatz eines Anwendungssystems entschieden (vgl. Krcmar /Informationsmanagement/ 152f.; Linthicum /EAI/ xvii).

¹¹² Vgl. Ließmann /EAI/ 180.

¹¹³ Brodie und Stonebraker berichten diesbezüglich von einer Fallstudie, in der sie in einem Unternehmen untersuchten, wie viele Anwendungssysteme von dem Betrieb einer zentralen Altanwendung abhingen. Dabei identifizierten sie unerwartete Abhängigkeiten zu ca. 1200 unterschiedlich komplexen und geschäftskritischen Anwendungssystemen (vgl. Brodie, Stonebraker /Legacy Systems/ 10). Zu den Interdependenzen zwischen verschiedenen betrieblichen Anwendungssystemen vgl. im Folgenden Lublinsky /EAI Implementation/ 26f.

Es muss ferner beachtet werden, dass sich die Abhängigkeiten zwischen den Anwendungssystemen im Zeitablauf verändern können. Dies kann aus Änderungen der Geschäftsstrategie, der zu unterstützenden Geschäftsprozesse oder der in den Unternehmen eingesetzten Anwendungssystemen resultieren. I. d. R. ist es nicht möglich, die einzelnen Anwendungssysteme in ausreichendem Maße an eine solche geänderte betriebliche Realität anzupassen, um Veränderungen der Abhängigkeiten zu vermeiden.

Die Abhängigkeiten zwischen Anwendungssystemen können in Anlehnung an Linß sowie an und Schumann anhand ihres Gegenstands, ihrer Richtung und ihrer Reichweite charakterisiert werden.¹¹⁴

▪ **Gegenstand der Abhängigkeiten**

Der Gegenstand einer Abhängigkeit bezeichnet, auf welche Aspekte der Informationsverarbeitung sie sich bezieht. Diesbezüglich erscheinen insbesondere die Informationen, die Aktivitäten und die Ablauflogik von Bedeutung. Zwischen Anwendungssystemen können mehrere Abhängigkeiten bestehen, die sich auf mehrere dieser Gegenstände beziehen können.

Eine Abhängigkeit zwischen zwei Anwendungssystemen bezüglich der Informationen bedeutet beispielsweise, dass für den Einsatz eines Anwendungssystems die Daten des anderen Anwendungssystems erforderlich sind. Sie kann aber auch darauf beruhen, dass eine Änderung der Daten eines Anwendungssystems eine Änderung der Daten des anderen Systems erforderlich macht.

Eine Abhängigkeit zwischen zwei Anwendungssystemen bezüglich der Aktivitäten bedeutet, dass im Zuge des Einsatzes eines Anwendungssystems Leistungen nachgefragt werden, die durch die Funktionen des anderen Anwendungssystems erbracht werden. Sie kann aber auch darauf beruhen, dass eine Aktivität durch Funktionen in beiden Anwendungssystemen realisiert wird, so dass die Konsistenz beider Realisierungen sichergestellt werden muss.

¹¹⁴ Vgl. Linß /Nutzeffekte/ 17-27; Schumann /Großintegrierte Informationsverarbeitung/ 6. Die Unterscheidung dieser Dimensionen der Abhängigkeiten zwischen Anwendungssystemen erfolgt in Anlehnung an das Konzept von Linß zur Beschreibung von Integrationszuständen (vgl. Linß /Nutzeffekte/ 17-26). Da ein Integrationsproblem als eine mangelnde Kongruenz von Integrationsbedarf und Integrationszustand verstanden wird, erscheint es sinnvoll beide Konzepte anhand der gleichen Dimensionen zu beschreiben.

Eine Abhängigkeit zwischen zwei Anwendungssystemen bezüglich der Ablauflogik bedeutet, dass die beiden Anwendungssysteme bei der Abwicklung eines Geschäftsprozesses oder mehrerer zusammenhängender Geschäftsprozesse in einer bestimmten Abfolge genutzt werden müssen.

▪ **Richtung der Abhängigkeiten**

Die Richtung der Abhängigkeiten bezeichnet, welche organisatorischen Ebenen sie überspannt. Gemäß Mertens können Anwendungssysteme anhand der durch sie unterstützten Informationsverarbeitungsaufgaben in operative Systeme, d. h. Administrations- oder Dispositionssysteme, sowie Planungs- und Kontrollsysteme unterschieden werden.¹¹⁵ Sie können darauf hin verschiedenen organisatorischen Ebenen zugeordnet werden, wobei die Planungs- und Kontrollsysteme prinzipiell über den Dispositionssystemen und diese wiederum über den Administrationssystemen angeordnet werden.

Ausgehend von diesem Verständnis können vertikale und horizontale Abhängigkeiten bestehen, wie Abbildung 2-2 in Form einer Pyramide veranschaulicht.¹¹⁶ Vertikale Abhängigkeiten bestehen zwischen Anwendungssystemen verschiedener organisatorischer Ebenen. Beispielsweise sind Planungs- und Kontrollsysteme oftmals auf Daten aus Operativen Systemen angewiesen. Dagegen bestehen horizontale Abhängigkeiten zwischen Anwendungssystemen derselben organisatorischen Ebene, die verschiedene Stufen der betrieblichen Wertschöpfungskette bzw. der Abwicklung von Geschäftsprozessen unterstützen. Dies gilt insbesondere für Operative Systeme. Horizontale Abhängigkeiten können jedoch auch auf höheren Schichten der Pyramide von Bedeutung sein, z. B. bei der Abstimmung von Ertrags-, Absatz- und Produktionsplanung.¹¹⁷

¹¹⁵ Vgl. im Folgenden Mertens /Integrierte Informationsverarbeitung/ 11-13. Andere Autoren nehmen eine abweichende Einteilung von Anwendungssystemen vor. So unterscheiden Stahlknecht und Hasenkamp beispielsweise verschiedene Ausprägungen von Administrations- und Dispositionssystemen, Führungssystemen und Querschnittssystemen (siehe Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 344-351).

¹¹⁶ Vgl. Kaib /EAI/ 19; Mertens /Integrierte Informationsverarbeitung/ 4-6.

¹¹⁷ Vgl. Mertens /Integrierte Informationsverarbeitung/ 4-6.

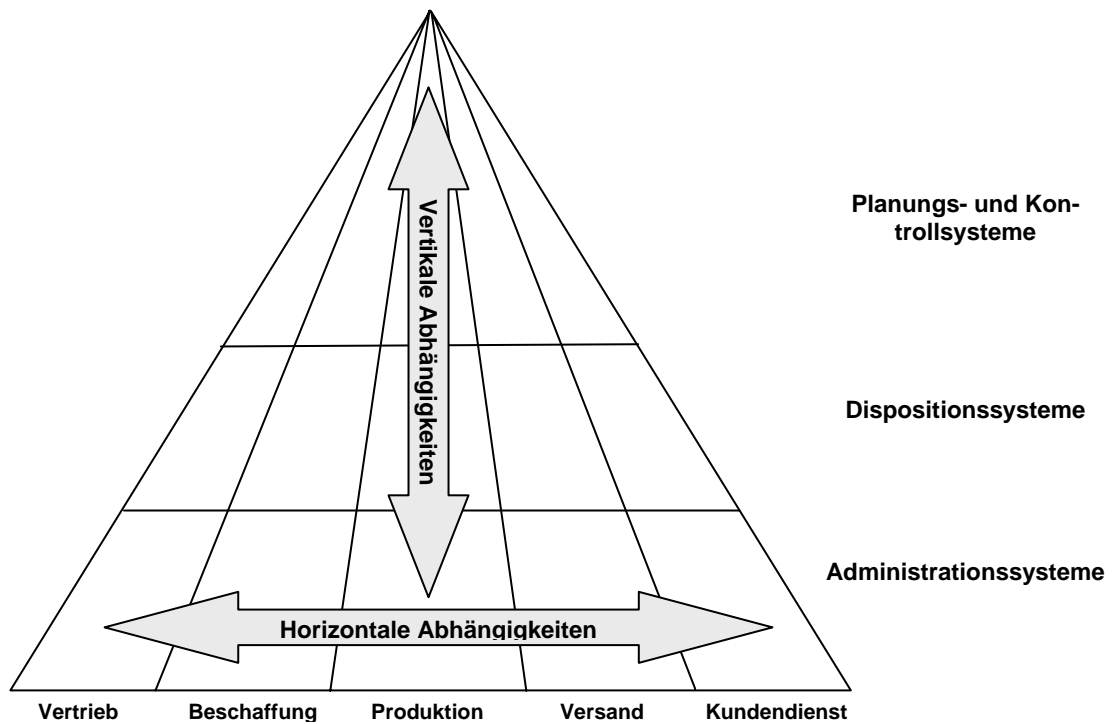


Abbildung 2-2: Richtung der Abhängigkeiten zwischen Anwendungssystemen¹¹⁸

▪ Reichweite der Abhängigkeiten

Die Reichweite einer Abhängigkeit bezeichnet, welche organisatorischen Grenzen sie überspannt.¹¹⁹ Diesbezüglich können innerbetriebliche und die zwischenbetriebliche Abhängigkeiten unterschieden werden.¹²⁰

Innerbetriebliche Abhängigkeiten betreffen Anwendungssysteme innerhalb eines Unternehmens. Sie können weiter daran charakterisiert werden, ob sie die Anwendungssysteme innerhalb eines Unternehmensbereichs oder die Anwendungssysteme mehrerer Unternehmensbereiche betreffen.

Zwischenbetriebliche Abhängigkeiten betreffen Anwendungssysteme mehrerer rechtlich selbständiger Unternehmen, die in wirtschaftlicher Beziehung zueinander stehen.

2.1.3 Hintergrund eines Integrationsbedarfs

Trotz bestehender Abhängigkeiten werden die Anwendungssysteme häufig getrennt von einander eingesetzt. Sie stellen damit „Informationsinseln“¹²¹ oder „Automatisierungs-

¹¹⁸ In Anlehnung an Mertens /Integrierte Informationsverarbeitung/ 5.

¹¹⁹ Vgl. Linß /Nutzeffekte/ 25.

¹²⁰ Vgl. Kaib /EAI/ 20.

inseln“¹²² in der betrieblichen Informationsverarbeitung dar. Als Folge müssen die Abhängigkeiten zwischen den Anwendungssystemen im Rahmen der Abwicklung der Geschäftsprozesse insbesondere durch die Benutzer¹²³ kompensiert werden. Dies ist i. d. R. mit verschiedenen Unzulänglichkeiten für die Unternehmen verbunden.¹²⁴

Beispielsweise ist eine manuelle Übertragung von Daten zwischen Anwendungssystemen, die wiederholte Eingabe und Pflege gleicher Daten in verschiedenen Anwendungssystemen sowie die wiederholte Realisierung und Pflege von geschäftlicher Logik in den Funktionen verschiedener Anwendungssysteme mit einem hohen Aufwand verbunden. Zudem stellen falsche oder unterlassene Eingaben und Pflegemaßnahmen ein hohes Risiko für die Korrektheit und Konsistenz redundanter Daten und Funktionen dar. Neben dem Aufwand und der Fehleranfälligkeit sind manuelle Arbeitsschritte zudem mit zeitlichen Verzögerungen verbunden. Diese können die Abwicklung von Geschäftsprozessen signifikant stören und die erfolgreiche Umsetzung von Geschäftsstrategien verhindern. Dies wird beispielsweise im Hinblick auf den elektronischen Handel und die dabei bestehenden hohen Anforderungen an die Geschwindigkeit der Informationsverarbeitung deutlich.¹²⁵

Schließlich bedingt der erforderliche Umgang der Benutzer mit den verschiedenen Anwendungssystemen einen hohen Einarbeitungsaufwand. Das Wissen darum, wann welches Anwendungssystem in welcher Weise einzusetzen ist, muss von den zuständigen Benutzern individuell erlernt werden. Dies wird durch die Spezifika der Benutzerführung der einzelnen Anwendungssysteme zusätzlich erschwert.

Wenn stattdessen die betroffenen Anwendungssysteme in geeigneter Weise zusammengeführt werden, kann dies zu positiven Effekten für die Unternehmen führen.¹²⁶ Diese Nutzeffekte können in operative und strategische Nutzeffekte unterteilt werden.¹²⁷

¹²¹ In der angloamerikanischen Literatur „islands of information“ oder auch „islands of data“ genannt (vgl. z. B. Lublinsky /EAI Implementation/ 26).

¹²² Vgl. Ferstl, Sinz /Wirtschaftsinformatik/ 216f. In der angloamerikanischen Literatur „islands of automation“ genannt (vgl. z. B. Lublinsky /EAI Implementation/ 26).

¹²³ Der Begriff des Benutzers bezeichnet eine einzelne Person oder eine Personengruppe, welche mit einem Anwendungssystem interagiert und dieses zur Erfüllung ihrer speziellen Aufgaben einsetzt. Demgegenüber werden die organisatorischen Einheiten, denen die Personen zugehören als Anwender bezeichnet. In diesem Sinne kann ein Anwender mehrere verschiedene Benutzer umfassen (vgl. Seibt /Anwender/ 43).

¹²⁴ Vgl. im Folgenden z. B. Kurbel, Rautenstrauch /Integration Engineering/ 168 oder Lublinsky /EAI Implementation/ 26f.

¹²⁵ Vgl. Kaib /EAI/ 43-52.

¹²⁶ Vgl. Linß /Nutzeffekte/ 30.

Operative Nutzeffekte betreffen die Informationsverarbeitung im Rahmen der bestehenden Geschäftsprozesse. Sie hängen wesentlich von den Charakteristika der betroffenen Geschäftsprozesse und den Beziehungen zwischen ihnen ab. Operative Nutzeffekte wirken sich damit in unterschiedlicher Weise in den Unternehmen aus, d. h. sie können verschiedenen Wirkungsarten zugeordnet werden. Beispielsweise können Kosten gespart, die Produktivität gesteigert, die Qualität und Flexibilität verbessert und die Sicherheit erhöht werden. Die Wirkungsarten operativer Nutzeffekte sind beispielhaft in Tabelle 2-1 dargestellt.¹²⁸

Wirkungsart	Erläuterung
Kosten	Nutzeffekte können sich beispielsweise in Kosteneinsparungen infolge eines geringeren Aufwands, einer höheren Qualität oder höheren Geschwindigkeit der Informationsverarbeitung bei der Abwicklung der Geschäftsprozesse ausdrücken. So sind beispielsweise Einsparungen bei Personalkosten aber auch bei Lagerhaltungs- oder Kapitalbindungskosten möglich. Nutzeffekte, die dieser Wirkungsart angehören, werden insbesondere für Wirtschaftlichkeitsbetrachtungen verwendet.
Produktivität	Nutzeffekte aufgrund einer verbesserten operativen Unterstützung der Geschäftsprozesse können zum einen die Menge der erzeugten Leistungen betreffen, wie z. B. eine höhere Ausstoßmenge an Produkten. Zum anderen können sie die dafür benötigte Zeit betreffen, wie z. B. Zeiteinsparungen in der Auftragsannahme.
Qualität	Nutzeffekte können die Qualität der im Rahmen der Geschäftsprozesse getroffenen Entscheidungen oder der erzeugten Leistungen betreffen. Ein Beispiel dafür ist eine bessere Kundenberatung aufgrund reichhaltiger, aktueller und korrekter Informationen.
Reaktionsfähigkeit	Nutzeffekte können die Reaktionsfähigkeit auf geänderte Bedingungen infolge der verbesserten Qualität und Geschwindigkeit der Informationsverarbeitung betreffen. Ein Beispiel dafür sind kurzfristige Änderungen im Produktionsprogramm aufgrund einer veränderten Nachfragesituation.
Sicherheit	Nutzeffekte können die Sicherheit der Informationsverarbeitung infolge einer Verringerung des Bedarfs und der Möglichkeit menschlicher Interventionen betreffen.

Tabelle 2-1: Beispielhafte Wirkungsarten operativer Nutzeffekte der Integration von Anwendungssystemen¹²⁹

¹²⁷ Für eine differenziertere Darstellung der operativen und strategischen Nutzenpotenziale infolge der Aufhebung der Isolation abhängiger Anwendungssysteme siehe beispielsweise Kaib /EAI/ 25-34.

¹²⁸ Die Bedeutung konkreter Nutzeffekte kann lediglich im Kontext der jeweiligen Unternehmen beurteilt werden. Sie sind in unterschiedlicher Weise monetär bewertbar, nur eingeschränkt vergleichbar und stehen teilweise in Wechselwirkung zueinander. Eine weiterführende Erläuterung und Untersuchung möglicher Nutzeffekte der Integration von Anwendungssystemen erfolgt beispielsweise in Linß /Nutzeffekte/ oder in Kaib /EAI/ 25-43.

¹²⁹ In Anlehnung an Linß /Nutzeffekte/ 35.

Strategische Nutzeffekte eröffnen neue unternehmerische Potenziale.¹³⁰ Beispielsweise können die Effektivität und Effizienz sowie die Steuerbarkeit der innerbetrieblichen und zwischenbetrieblichen Geschäftsprozesse verbessert werden und damit die Reaktionsfähigkeit der Unternehmen auf sich ändernde Wettbewerbsbedingungen verbessert werden.¹³¹ Einer geeigneten Integration der Anwendungssysteme wird ferner entscheidende Bedeutung für die erfolgreiche Eingliederung von Unternehmen bzw. abgegrenzten größeren Teilen von Unternehmen infolge eines Zusammenschlusses oder Erwerbs (sog. Mergers & Aquisitions) beigemessen.¹³²

Wird mit einer Integration von Anwendungssystemen aufgrund der zwischen ihnen bestehenden Abhängigkeiten eine insgesamt positive Wirkung für die Unternehmen erwartet, so wird dies im Weiteren als *Integrationsbedarf* bezeichnet. In dem Maße, in dem sich die Abhängigkeiten zwischen den Anwendungssystemen oder die angestrebten Nutzeffekte einer Integration im Zeitablauf verändern, ändert sich auch der bestehende Integrationsbedarf.

2.1.4 Dimensionen eines Integrationszustands

Ein Integrationszustand von Anwendungssystemen beschreibt, inwiefern die Anwendungssysteme miteinander integriert, d. h. zusammengeführt, sind. In diesem Sinne stellt auch der extreme Fall, dass keinerlei Verbindungen zwischen den Anwendungssystemen bestehen, einen Integrationszustand dar. Sowohl der Ausgangspunkt als auch das Ergebnis der Anwendungsintegration repräsentieren Integrationszustände.¹³³

¹³⁰ Vgl. Kaib /EAI/ 25f., 30.

¹³¹ Vgl. z. B. Kirchmer /Standardsoftware/ 7-10, Holten /Integration/ 41, Davenport /Enterprise System/ 123 sowie Kaib /EAI/ 30-34.

¹³² Vgl. z. B. Kromer, Stucky /Mergers & Aquisitions/ 523 oder Trapp, Otto /EAI/ 102-104.

¹³³ In der deutschsprachigen Literatur der Wirtschaftsinformatik werden verschiedene Konzepte vorgeschlagen, um Integrationszustände der betrieblichen Informationsverarbeitung zu beschreiben. Einen Überblick über verschiedene Ansätze zur Beschreibung von Integrationszuständen bieten beispielsweise Linß (vgl. Linß /Nutzeffekte/ 7-17) sowie Mertens und Holzer (vgl. Mertens, Holzer /Integrationsansätze/ 5-24). Diese Konzepte gehen jedoch vielfach über den in dieser Arbeit betrachteten Integrationsansatz hinaus.

Die Abhängigkeiten zwischen Anwendungssystemen wurden anhand ihres Gegenstands, ihrer Richtung und ihrer Reichweite charakterisiert. Diese Dimensionen können auch zur Beschreibung eines Integrationszustands dienen.¹³⁴

▪ **Gegenstand der Integration**

Der Gegenstand der Integration bezeichnet, auf welche Aspekte der Anwendungssysteme sich die Integration bezieht, d. h. unter welchen Gesichtspunkten die Anwendungssysteme zusammengeführt werden sollen.¹³⁵ Bezugnehmend auf die Erläuterung der Abhängigkeiten zwischen den Anwendungssystemen erscheinen diesbezüglich insbesondere drei Gegenstände von Bedeutung: die Daten der Anwendungssysteme, die Funktionen der Anwendungssysteme und die Prozessabwicklung.

Die Integration der Daten bezieht sich auf die automatisierte Versorgung der Anwendungssysteme mit den für sie relevanten Daten sowie eine automatisierte Pflege der verteilten Datenbestände.

Die Integration der Funktionen bezieht sich auf die Verfügbarkeit von Funktionen über die Grenzen einzelner Anwendungssysteme hinweg.¹³⁶

Die Integration der Prozessabwicklung bezieht sich auf die durchgängige Unterstützung von mehrstufigen Geschäftsprozessen oder Geschäftsprozessfolgen.¹³⁷ Nicht allein die Versorgung mit Daten und deren Pflege oder die Verfügbarkeit von Funktionen, sondern deren koordinierter Einsatz zur automatisierten Abwicklung zusammenhängender Aktivitäten ist in diesem Sinne von Bedeutung. Die Integration der Prozessabwicklung ist somit komplementär zur Integration der Daten und der Integration der Funktionen.¹³⁸

▪ **Richtung der Integration**

Die Richtung der Integration bezeichnet, welche organisatorischen Ebenen mit der Zusammenführung der Anwendungssysteme überspannt werden. Wiederum analog zu den Abhängigkeiten können diesbezüglich eine vertikale und eine horizontale Dimension unterschieden werden.

¹³⁴ Vgl. Schumann /Großintegrierte Informationsverarbeitung/ 6; Linß /Nutzeffekte/ 17-27.

¹³⁵ Vgl. Linß /Nutzeffekte/ 19.

¹³⁶ Vgl. Boles, Friebe, Luhmann /Integrationsszenarien/ 58f.

¹³⁷ Vgl. Boles, Friebe, Luhmann /Integrationsszenarien/ 59.

¹³⁸ Vgl. Linthicum /Application Integration/ 56.

Die vertikale Integration bezieht sich auf die Zusammenführung von Anwendungssystemen verschiedener organisatorischer Ebenen, d. h. die Verknüpfung von Administrationssystemen, Dispositionssystemen und Planungs- oder Kontrollsystemen. Die horizontale Integration bezieht sich dagegen auf Anwendungssysteme derselben organisatorischen Ebene, die verschiedene Stufen der betrieblichen Wertschöpfungskette unterstützen, wie beispielsweise die Beschaffung, die Produktion oder den Vertrieb.¹³⁹

▪ **Reichweite der Integration**

Die Reichweite der Integration bezeichnet, über welche organisatorischen Grenzen hinweg Anwendungssysteme zusammengeführt werden.¹⁴⁰ Analog zur Erläuterung der Abhängigkeiten zwischen den Anwendungssystemen können die innerbetriebliche und die zwischenbetriebliche Integration unterschieden werden. Ebenfalls kann die innerbetriebliche Integration danach unterteilt werden, ob sie Anwendungssysteme eines Unternehmensbereichs oder mehrerer Unternehmensbereiche betrifft.¹⁴¹

Die innerbetriebliche und die zwischenbetriebliche Integration verhalten sich nicht alternativ, sondern vielmehr komplementär zueinander. Die innerbetriebliche Integration wird dabei häufig als notwendiger erster Schritt betrachtet, auf dem eine zwischenbetriebliche Integration aufbaut.¹⁴²

Als dritte Ausprägung der Integrationsreichweite wird zuweilen die Verknüpfung innerbetrieblicher Anwendungssysteme mit internetbasierten Anwendungssystemen angesehen, die durch die Kunden der Unternehmen genutzt werden. In diesem Zusammenhang wird beispielsweise von „Web Integration“, „Internet Application Integration“ oder „Business-to-Consumer (B2C) Integration“ gesprochen.¹⁴³ Themistocleous und Irani ordnen diese Ausprägung als „hybride Anwendungsintegration“ zwischen der innerbetrieblichen und zwischenbetrieblichen Integration ein.¹⁴⁴

¹³⁹ Vgl. Mertens /Integrierte Informationsverarbeitung/ 4-6; Losavio, Ortega, Pérez /EAI/ 197.

¹⁴⁰ Vgl. Linß /Nutzeffekte/ 25.

¹⁴¹ Vgl. Mertens /Integrierte Informationsverarbeitung/ 6.

¹⁴² Vgl. Linthicum /B2B Integration/ 16, Pinkston /Integration/ 52.

¹⁴³ Vgl. Lam /Enterprise Integration/ 292 oder Lee, Siau, Hong /Enterprise Integration/ 58f.

¹⁴⁴ Vgl. Themistocleous, Irani /Taxonomy for Application Integration/ 158-162.

▪ **Qualität als vierte Dimension eines Integrationszustands**

Qualität kann als vierte Dimension eines Integrationszustands angesehen werden, denn die zuvor erläuterten strukturellen Merkmale eines Integrationszustands können auf qualitativ unterschiedliche Weise realisiert werden.

Für die Bestimmung der Qualität von Softwaresystemen werden in der Fachliteratur verschiedene Merkmale verwendet. Diese Qualitätsmerkmale können in zwei Kategorien unterteilt werden: in solche, die im Rahmen der Ausführung bzw. dem Einsatz des Softwaresystems wahrgenommen werden können und solche, die nicht im Rahmen der Ausführung bzw. dem Einsatz des Softwaresystems wahrgenommen werden können.¹⁴⁵

Dieses Verständnis von Qualität kann auch auf Integrationszustände von Anwendungssystemen angewandt werden. Demnach kann ein Integrationszustand zum einen anhand von Qualitätsmerkmalen charakterisiert werden, welche im Rahmen des Einsatzes der Anwendungssysteme wahrgenommen werden können. Diesbezüglich können beispielsweise die Zuverlässigkeit, das Zeitverhalten oder die Sicherheit angeführt werden. Zum anderen kann ein Integrationszustand anhand von Qualitätsmerkmalen charakterisiert werden, die nicht im Rahmen des Einsatzes der Anwendungssysteme wahrgenommen werden können. Diesbezüglich erscheint beispielsweise die Flexibilität von Bedeutung. Prinzipiell gilt, dass eine möglichst hohe Ausprägung eines Qualitätsmerkmals erstrebenswert ist. Es können jedoch Konflikte hinsichtlich des Verfolgens verschiedener Qualitätsmerkmale bestehen.¹⁴⁶

Die genannten Qualitätsmerkmale zur Charakterisierung von Integrationszuständen werden in Tabelle 2-2 kurz dargestellt. Sie werden im Rahmen der Ausführungen zu Integrationslösungen nochmals aufgegriffen und ausführlicher erläutert.

¹⁴⁵ Vgl. beispielsweise Bass, Clements, Kazman /Software Architecture/ 76-85; DIN (Hrsg.) /66272/.

¹⁴⁶ Vgl. Bass, Clements, Kazman /Software Architecture/ 78.

Qualitätsmerkmal	Erläuterung
Zuverlässigkeit	Die Zuverlässigkeit eines Integrationszustandes bezeichnet die Fähigkeit, den fehlerfreien Betrieb im Zeitverlauf aufrecht zu erhalten. ¹⁴⁷ In engem Zusammenhang mit der Zuverlässigkeit stehen Verfügbarkeit und Wiederherstellbarkeit. ¹⁴⁸
Zeitverhalten	Das Zeitverhalten eines Integrationszustandes bezeichnet die Fähigkeit, die Integration der Daten, der Funktionen und der Nutzungsabfolge unter bestimmten Bedingungen mit angemessenen Latenzzeiten zu gewährleisten. ¹⁴⁹
Sicherheit	Die Sicherheit eines Integrationszustandes bezeichnet die Unterstützung eines konsistenten Sicherheitskonzeptes. ¹⁵⁰ Dieses umfasst beispielsweise Anforderungen an die Authentifizierung, die Autorisierung, die Kommunikationssicherheit oder die Nachvollziehbarkeit. ¹⁵¹
Flexibilität	Die Flexibilität eines Integrationszustands bezeichnet die Möglichkeit, mit geringem Aufwand eine Veränderung des Integrationszustands vornehmen zu können. Dies kann sowohl die strukturellen als auch die qualitativen Merkmale des Integrationszustands betreffen. ¹⁵²

Tabelle 2-2: Beispielhafte Qualitätsmerkmale zur Charakterisierung von Integrationszuständen

2.1.5 Integrationsprobleme als unzureichende Integrationszustände von Anwendungssystemen

Aufbauend auf den vorangehenden Ausführungen kann ein Integrationsproblem auf zwei Aspekten basieren. Zum einen darauf, dass ein Integrationszustand von Anwendungssystemen als strukturell unzureichend angesehen wird. Zum anderen darauf, dass ein Integrationszustand von Anwendungssystemen als qualitativ unzureichend angesehen wird.

Ein strukturell unzureichender Integrationszustand von Anwendungssystemen wird den gegebenen Abhängigkeiten zwischen ihnen im Hinblick auf Gegenstand, Richtung und/oder Reichweite nicht gerecht. Damit besteht wie beschrieben die Notwendigkeit, die Abhängigkeiten zwischen den Anwendungssystemen insbesondere durch die Benutzer zu kompensieren, d. h. ein bestehender Integrationsbedarf wird nicht abgedeckt.

Ein qualitativ unzureichender Integrationszustand von Anwendungssystemen bedeutet, dass die Ausprägungen der Qualitätsmerkmale des Integrationszustands verhindern, die angestrebten Nutzeffekte einer Integration auszuschöpfen. Beispielsweise können angestrebte operative Nutzeffekte im Hinblick auf die Kosten nur bei einem angemessenen

¹⁴⁷ Vgl. Bass, Clements, Kazman /Software Architecture/ 80.

¹⁴⁸ Vgl. DIN /66272/ 8.

¹⁴⁹ Vgl. Losavio, Ortega, Pérez /EAI Quality/ 122.

¹⁵⁰ Vgl. Andersson, Johnson /Integration Styles/ 227; Ruh, Maginnis, Brown /EAI/ 15.

¹⁵¹ Vgl. Cummins /Enterprise Integration/ 321-324.

¹⁵² Vgl. Kaib /EAI/ 82.

Zeitverhalten und strategische Nutzeffekte im Hinblick auf die Reaktionsfähigkeit der Unternehmen auf sich ändernde Wettbewerbsbedingungen nur bei einer angemessenen Flexibilität verwirklicht werden. In der Praxis haben sich die mit der Integration verbundenen Hoffnungen hinsichtlich der durch sie realisierbaren Nutzeffekte infolge qualitativer Mängel der realisierten Integrationszustände häufig nicht erfüllt.¹⁵³

Ein qualitativ unzureichender Integrationszustand kann darüber hinaus zu negativen Effekten führen, die über das Nichterreichen angestrebter Nutzeffekte hinausreichen. So kann ein Integrationszustand beispielsweise mit einem hohen Pflegeaufwand verbunden sein.¹⁵⁴ Die negativen Auswirkungen fehlerhafter Informationen können zudem potenziert werden, wenn sich diese unkontrolliert über die integrierten Anwendungssysteme fortpflanzen.¹⁵⁵

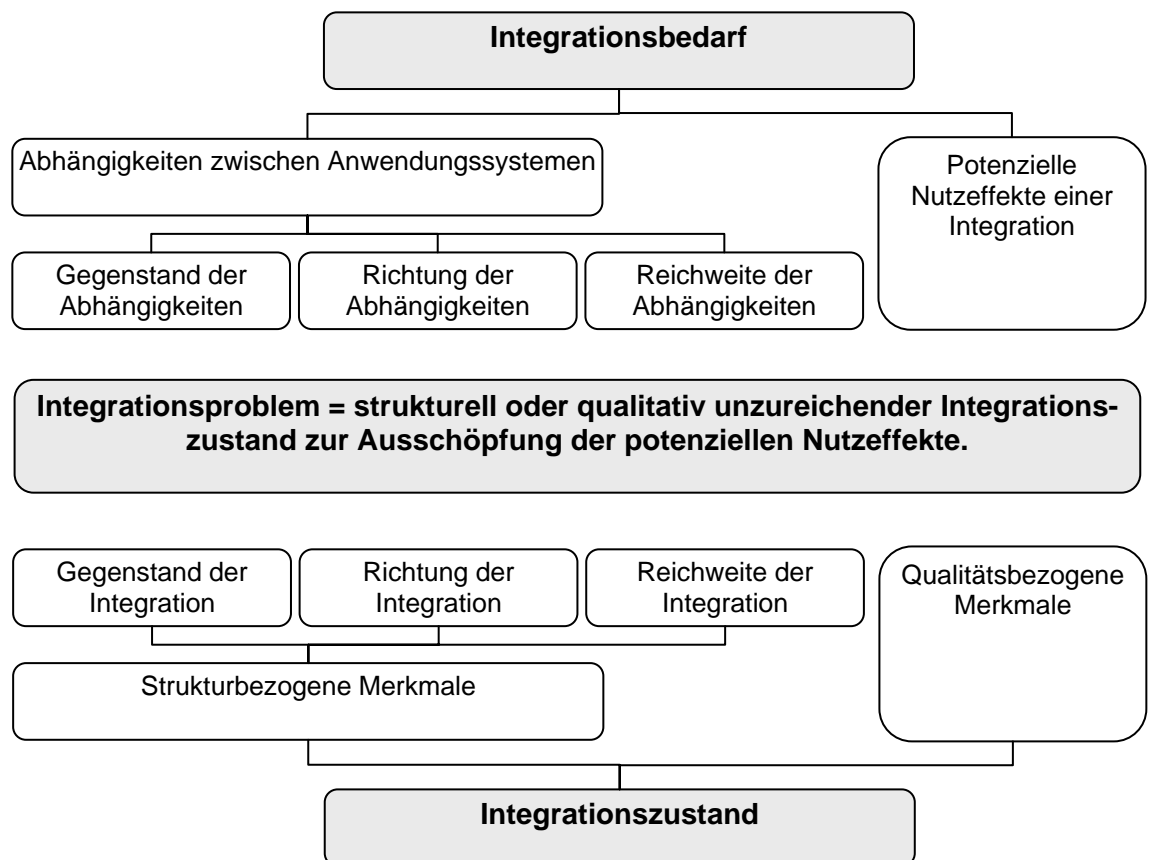


Abbildung 2-3: Integrationsproblem als fehlende Kongruenz von Integrationsbedarf und Integrationszustand

¹⁵³ Vgl. beispielsweise Kurbel, Rautenstrauch /Integration Engineering/ 168.

¹⁵⁴ Vgl. Kaib /EAI/ 28.

¹⁵⁵ Vgl. Mertens /Integrierte Informationsverarbeitung/ 10.

Das Nichterreichen potenzieller Nutzeffekte und die Unzulänglichkeiten bzw. negativen Effekte infolge strukturell und/oder qualitativ unzureichender Integrationszustände kann eine ernste Bedrohung für den wirtschaftlichen Erfolg der Unternehmen bedeuten.¹⁵⁶ Prinzipiell können zwei Wege zur Lösung eines solchen Integrationsproblems beschritten werden.

Zum einen kann der Integrationsbedarf verändert und den strukturellen und qualitativen Merkmalen des Integrationszustands angenähert werden. Ein Integrationsbedarf kann insbesondere durch Maßnahmen innerhalb der unterschiedenen Gestaltungsebenen verändert werden, d. h. durch Veränderungen der Strategie, der Geschäftsprozesse oder der Anwendungssysteme. Im Rahmen dieser Arbeit wird dieser Lösungsweg von Integrationsproblemen jedoch nicht betrachtet. Es wird vielmehr von einer gegebenen Strategie, gegebenen Geschäftsprozessen und gegebenen Anwendungssystemen der Unternehmen ausgegangen.

Zum anderen kann der Integrationszustand der Anwendungssysteme verändert und den strukturellen und qualitativen Merkmalen des Integrationsbedarfs angeglichen werden. Dies ist der Lösungsraum, der in dieser Arbeit im Hinblick auf den Integrationsansatz der Anwendungsintegration betrachtet wird.¹⁵⁷

2.1.6 Charakteristika der Anwendungssysteme

2.1.6.1 Konzeptionelle und technische Merkmale der Anwendungssysteme

Ein Integrationsproblem wird durch die konzeptionellen und technischen Merkmale der zu verknüpfenden Anwendungssysteme geprägt. Diese beeinflussen nicht nur den bestehenden Integrationsbedarf, sie begründen auch Restriktionen für die Integrationslösung.

Darüber hinaus müssen oftmals auch die konzeptionellen und technischen Merkmale anderer Maßnahmen zur Integration der Anwendungssysteme berücksichtigt werden, die verschiedene der eingangs erläuterten Integrationsansätze verfolgen können. Auf die Merkmale von Integrationslösungen wird im Folgenden jedoch nicht eingegangen. Sie werden in Kapitel 2.2 detailliert behandelt.

¹⁵⁶ Für eine einführende Darstellung der Bedeutung der Anwendungssysteme in den Unternehmen siehe z. B. Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 344-451.

¹⁵⁷ In der Praxis werden die beiden Wege häufig kombiniert bei der Lösung eines Integrationsproblems begangen. Die Integration der Anwendungssysteme erfolgt dann gemeinsam mit Änderungen der anderen unterschiedenen Gestaltungsebenen, um eine umfassende Restrukturierung, Modernisierung und Konsolidierung der betrieblichen Informationsverarbeitung vorzunehmen.

▪ **Konzeptionelle Merkmale der Anwendungssysteme**

Bei der Integration von Anwendungssystemen müssen zum einen ihre konzeptionellen Charakteristika berücksichtigt werden. In ihnen kommt die spezifische Abbildung des zu unterstützenden Ausschnitts der betrieblichen Informationsverarbeitung zum Ausdruck. Von den bereits erwähnten Daten-, Funktions- und Kontrollflussmodellen werden beispielhaft die Datenmodelle näher erläutert.

Datenmodelle beschreiben ein spezifisches Datenschema durch die Abgrenzung und Spezifikation von Datenelementen sowie der Definition der Beziehungen zwischen den Datenelementen.¹⁵⁸ Für die Abgrenzung der Datenelemente müssen diese mit einem eindeutigen Namen bezeichnet werden, über den sie identifiziert werden können.¹⁵⁹ Die Spezifikation der Datenelemente erfolgt beispielsweise durch das Festlegen des Definitionsbereichs (im Sinne der Festlegung des Datentyps sowie der Einschränkung der Werte, welche das Datenelement annehmen kann) und des Formats des Datenelements.¹⁶⁰ Die definierten Beziehungen zwischen den Datenelementen bilden die Abhängigkeiten zwischen den durch die Datenelemente beschriebenen Entitäten ab. Sie können um ergänzende Randbedingungen (engl. constraints) ergänzt werden.¹⁶¹ Zusammenfassend betrifft ein Datenmodell somit zwei Aspekte. Es definiert die Syntax der Daten, in dem Sinne, dass es die formale Beschreibung der Daten festlegt. Es basiert darüber hinaus auf einer konkreten Semantik der Daten in dem Sinne, dass ihm ein konkretes Verständnis der abgebildeten Realitätsausschnitte zugrunde liegt.

▪ **Technische Merkmale der Anwendungssysteme**

Zum anderen sind die technischen Charakteristika der Anwendungssysteme für ihre Integration von Bedeutung. Dies betrifft insbesondere die Merkmale, welche die Interoperabilität der Anwendungssysteme determinieren, d. h. ihre Eignung, mit anderen Systemen zu interagieren.¹⁶² Relevant erscheinen jedoch auch solche Merkmale, welche die Änderbarkeit oder Erweiterbarkeit der Anwendungssysteme betreffen. Sie begrün-

¹⁵⁸ Vgl. Elmasri, Navathe /Database Systems/ 27f.

¹⁵⁹ Vgl. Wiederhold, Quian /Database/ 271.

¹⁶⁰ Vgl. Wiederhold, Quian /Database/ 272.

¹⁶¹ Vgl. Wiederhold, Quian /Database/ 272f.; Morris u. a. /Database/ 288.

¹⁶² Vgl. Bass, Clements, Kazman /Software Architecture/ 85.

den beispielsweise, inwiefern die Interoperabilität der Anwendungssysteme durch Eingriffe verbessert werden kann.

Im Hinblick auf die technischen Charakteristika der Anwendungssysteme wird in der Fachliteratur insbesondere auf ihre externen Schnittstellen, ihre interne Schichtentrennung und die Datenunabhängigkeit hingewiesen.

Schnittstellen bezeichnen im Allgemeinen Interaktionspunkte an der Grenze zweier Systeme, mit vereinbarten Regeln, nach denen die beiden Systeme in Wechselwirkung zueinander treten können.¹⁶³ Die externen Schnittstellen der Anwendungssysteme bezeichnen in diesem Sinne verfügbare Interaktionspunkte zwischen den Anwendungssystemen und anderen Softwaresystemen.

Ein Anwendungssystem kann mehrere externe Schnittstellen besitzen, die sich durch spezifische Eigenschaften auszeichnen.¹⁶⁴ Zur Beschreibung dieser Eigenschaften können verschiedene Merkmale unterschieden werden, die beispielhaft in Tabelle 2-3 erläutert werden.

Merkmal	Erläuterung
Ausrichtung	Es können Exportschnittstellen und Importschnittstellen unterschieden werden. ¹⁶⁵ Exportschnittstellen ermöglichen externen Softwaresystemen in spezifischer Weise den Zugriff auf Funktionen oder Daten eines Anwendungssystems. ¹⁶⁶ Importschnittstellen beschreiben dagegen die Möglichkeiten eines Anwendungssystems die Exportschnittstellen anderer Softwaresysteme zu nutzen. ¹⁶⁷
Gegenstand	Externe Schnittstellen auf der Ebene des Datenmodells betreffen z. B. den externen Zugriff auf Daten eines Anwendungssystems, d. h. deren Abfrage oder deren Manipulation. Letzteres kann mit oder ohne Integritäts- und Plausibilitätsprüfungen erfolgen. ¹⁶⁸ Externe Schnittstellen auf der Ebene des Funktionsmodells betreffen z. B. den externen Zugriff auf Fachfunktionalität des Anwendungssystems. Solche Exportschnittstellen können nicht nur der Zugriff auf Daten ermöglichen sondern definierter Dienste für andere Softwaresysteme bereitstellen. ¹⁶⁹

¹⁶³ Vgl. Clements u. a. /Documenting Architectures/ 223.

¹⁶⁴ Vgl. Clements u. a. /Documenting Architectures/ 225.

¹⁶⁵ Vgl. Ließmann /Schnittstellenorientierung/ 7.

¹⁶⁶ Vgl. Linthicum /EAI/ 39; Ließmann /Schnittstellenorientierung/ 7.

¹⁶⁷ Vgl. Ließmann /Schnittstellenorientierung/ 7.

¹⁶⁸ Vgl. Linthicum /EAI/ 49.

¹⁶⁹ Vgl. Linthicum /Adapters/ 40.

Merkmal	Erläuterung
Variabilität	Externe Schnittstellen können entweder fest definierte Zugriffsmöglichkeiten schaffen oder eine variable Bestimmung der Daten oder Funktionen ermöglichen, auf die zugegriffen werden soll. ¹⁷⁰ Beispielsweise ermöglichen die Programmierschnittstellen moderner Anwendungssysteme (engl. application programming interface, API) oftmals eine hohe Variabilität. ¹⁷¹
Interaktionsmechanismus	Die Interaktion über externe Schnittstellen kann mittels Dateien, Nachrichten, Prozeduraufrufen oder verteilten Objekten erfolgen. ¹⁷²
Kommunikationsregeln	Die externen Schnittstellen definieren Regeln, nach denen eine Kommunikation erfolgen kann und legen damit Anforderungen an potenzielle Interaktionspartner fest. Diese Regeln betreffen beispielsweise den Aufbau und Abbau von Verbindungen, die Form der Anfragen und Antworten, die über die Schnittstelle ausgetauscht werden, das zugrunde liegende Kommunikationsmodell, die unterstützten Protokolle und Programmiersprachen oder sonstige Restriktionen. Die externen Schnittstellen zeichnen sich in diesem Zusammenhang auch durch spezifische Fähigkeiten zur Lokalisierung von Interaktionspartnern oder die Prüfung deren Verfügbarkeit aus. ¹⁷³
Fehlertoleranz	Externe Schnittstellen bieten unterschiedliche Mechanismen zum Umgang mit Fehlern, die im Rahmen der Interaktion auftreten, beispielsweise weil Ereignisse eintreten, die den der Schnittstelle zugrunde liegenden Annahmen widersprechen. ¹⁷⁴
Standardisierung	Die externen Schnittstellen der Anwendungssysteme unterstützen in unterschiedlicher Weise Standards. Während die externen Schnittstellen der Anwendungssysteme früher häufig proprietär gestaltet wurden (dies gilt also vorwiegend für Altsysteme) unterstützen die externen Schnittstellen moderner Anwendungssysteme häufig bestimmte Standards (z. B. RMI, CORBA, DCOM). ¹⁷⁵
Qualitätsmerkmale	Die externen Schnittstellen der Anwendungssysteme zeichnen sich im Zuge einer Interaktion durch spezifische Ausprägungen von Qualitätsmerkmalen, wie der Performanz, der Zuverlässigkeit oder der Verfügbarkeit aus. ¹⁷⁶

Tabelle 2-3: Beispiele für Eigenschaften externer Schnittstellen der Anwendungssysteme

¹⁷⁰ Vgl. Linthicum /Adapters/ 40.

¹⁷¹ Vgl. Linthicum /EAI/ 39.

¹⁷² Vgl. Ließmann /Schnittstellenorientierung/ 2; Goodyear /Enterprise System Architecture/ Kapitel 3; S. 27.

¹⁷³ Vgl. Andersson, Johnson /Integration Styles/ 227 Einen Eindruck von der technischen Vielfalt von Schnittstellen vermittelt beispielsweise Mehta, Medvidovic, Phadke /Software Connectors/ 182.

¹⁷⁴ Vgl. Clements u. a. /Documenting Architectures/ 231-237.

¹⁷⁵ Vgl. Linthicum /EAI/ 38.

¹⁷⁶ Vgl. Clements u. a. /Documenting Architectures/ 232.

Die Anzahl und die Eigenschaften der externen Schnittstellen eines Anwendungssystems stehen i. d. R. im Zusammenhang mit dem Ausmaß der internen Schichtentrennung der Anwendungssysteme.¹⁷⁷

In Anlehnung an das Client/Server-Architekturmodell können die Anwendungssysteme idealtypisch in die drei Schichten Benutzerschnittstelle,¹⁷⁸ Anwendungsfunktionalität¹⁷⁹ und Datenverwaltung¹⁸⁰ unterteilt werden.¹⁸¹

Das Ausmaß der Trennung dieser Schichten kommt zum einen darin zum Ausdruck, inwiefern sie durch verschiedene Softwarekomponenten realisiert werden,¹⁸² und zum anderen in der Stärke der Kopplung dieser Softwarekomponenten.¹⁸³ Es können beispielsweise Einschicht-Architekturen, Zweischicht-Architekturen oder Dreischicht-Architekturen unterschieden werden.¹⁸⁴

¹⁷⁷ Vgl. Brodie, Stonebraker /Legacy Systems/ 17; Keller /EAI/ 186.

¹⁷⁸ Die Benutzerschnittstelle stellt das Bindeglied zwischen einem Benutzer und dem Anwendungssystem dar. Sie ist für die Ein- und Ausgabe von Informationen sowie die Übermittlung von Aufrufen von Funktionen im Rahmen der Interaktion mit dem Benutzer zuständig (vgl. Riehm, Vogler /Middleware/ 29; Chari, Seshadri /Integration/ 61).

¹⁷⁹ Die Anwendungsfunktionalität beinhaltet zum einen die Funktionen zur Unterstützung der betrieblichen Aktivitäten. Darüber hinaus umfasst sie die mit der Präsentations- und der Datenzugriffsfunktionalität Funktionen zur Steuerung der über- und untergeordneten Ebenen (vgl. Riehm, Vogler /Middleware/ 29; Chari, Seshadri /Integration/ 61; Ruh, Maginnis, Brown /EAI/ 27).

¹⁸⁰ Die Datenverwaltung umfasst die Komponenten der Anwendungssysteme, welche das Speichern und den Zugriff von Daten sowie die Verwaltung der Datenbestände unterstützen. Die Datenverwaltungsschicht zeichnet sich ferner durch spezifische Methoden zur Unterstützung des Datenschutzes, der Datensicherheit und der Datenintegrität sowie durch ein unterschiedliches Maß an Redundanz aus (vgl. Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 208f.).

¹⁸¹ Vgl. Riehm, Vogler /Middleware/ 29; Goodyear /Enterprise System Architecture/ Kapitel 1, S. 15-18.

¹⁸² Die einzelnen Schichten der Anwendungssoftware können teilweise durch selbständig lauffähige Softwaresysteme gebildet werden. Dies wird insbesondere im Hinblick auf die Datenverwaltung deutlich. Im Laufe der vergangenen Dekaden wurden in zunehmendem Maße Funktionen der Datenverwaltung gebündelt und aus den Anwendungssystemen in Systemsoftware und systemnahe Software verlagert. Während die Systemsoftware die Steuerung der physischen Datenhaltung in Verbindung mit Hardwarekomponenten durchführt, decken vielfach spezielle Datenbanksysteme als systemnahe Software die Funktionalität der logischen Datenverwaltung ab (vgl. Wedekind /Datenbanksystem/ 139f. Zum Begriff der Datenbanksysteme vgl. Elmasri, Navathe /Database Systems/ 4f.).

¹⁸³ Unter Kopplung wird in diesem Kontext die Bindung zwischen den verschiedenen Komponenten verstanden, die auch als Adhäsion bezeichnet wird (vgl. Heyliger /Coupling/ 220). Je enger die Kopplung, desto geringer die Eigenständigkeit der Komponenten. Dies bedeutet, dass Implementierungsabhängigkeiten zwischen den Anwendungssystemen bestehen und Modifikationen an einer Komponente Anpassungen an den anderen gekoppelten Komponenten bedingen (vgl. Kaib /EAI/ 16; Ruh, Maginnis, Brown /EAI/ 21).

¹⁸⁴ Vgl. Linthicum /EAI/ 44-47. Das Client/Server-Architekturmodell der Gartner Group unterscheidet weiterreichend fünf verschiedene Ausprägungen der Architekturen hinsichtlich der beschriebenen Schichten (vgl. Goodyear /Enterprise System Architecture/ Kapitel 1, S. 15-18).

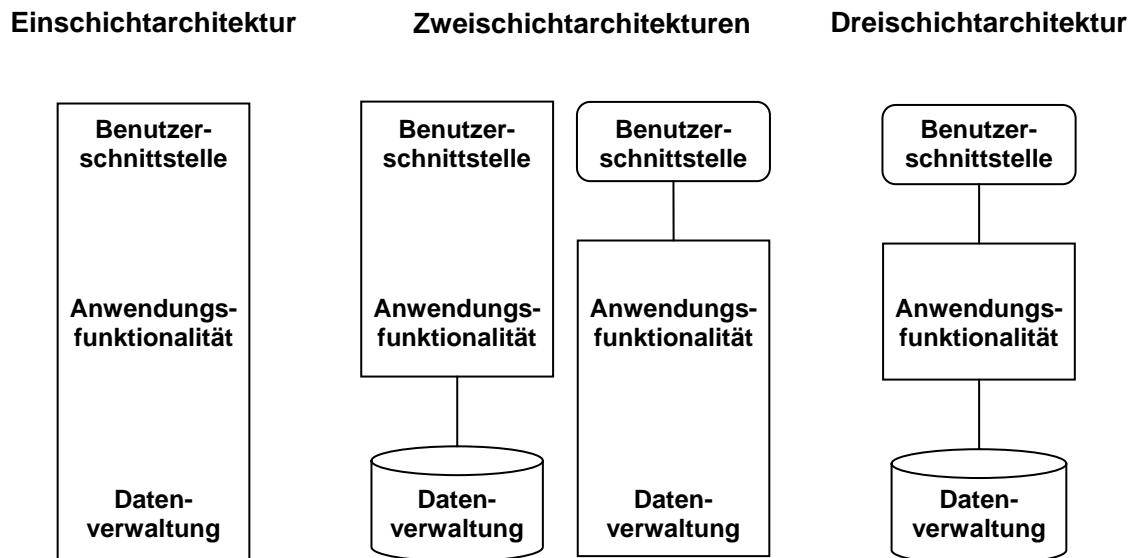


Abbildung 2-4: Einschicht-, Zweischicht- und Dreischichtarchitekturen

Für die Integration von Anwendungssystemen erscheint ferner das Konzept der Datenunabhängigkeit von Bedeutung. Dieses umfasst zwei Bereiche.

Zum einen betrifft die Datenunabhängigkeit die Unabhängigkeit der physischen von der logischen Datenorganisation (sog. physische Datenunabhängigkeit), d. h. es erfolgt eine Trennung eines konzeptionellen Datenmodells, welches die Struktur der Daten im Hinblick auf die durch sie beschriebenen Entitäten beschreibt und somit eine präzise Definition der Semantik des abzubildenden Realitätsausschnitts anstrebt (logische Datenorganisation oder konzeptionelles Schema), von den Details der physischen Datenspeicherung und den Methoden der Manipulation des Datenbestands (physische Datenorganisation oder internes Schema).¹⁸⁵

Zum anderen betrifft die Datenunabhängigkeit die Unabhängigkeit der gespeicherten Daten von der Anwendungssoftware. Dies bedeutet, dass die Speicherung der Daten anwendungsneutral, d. h. technisch unabhängig von der erzeugenden oder benutzenden Anwendungssoftware gespeichert sind.¹⁸⁶

In der Geschichte der Datenverarbeitung sind nacheinander verschiedene Formen der Datenhaltung entstanden, die zu einem unterschiedlichen Ausmaß an Datenunabhängigkeit führen.¹⁸⁷

¹⁸⁵ Vgl. Elmasri, Navathe /Database Systems/ 29-32; Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 207; Falkenberg /Datenstruktur/ 212.

¹⁸⁶ Vgl. Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 207.

¹⁸⁷ Zunächst erfolgte die Datenhaltung vorwiegend in Form der dateibasierten Datenverwaltung bei der die Daten sequenziell satzweise abgelegt werden. Diese zeichnet sich insbesondere durch eine gerin-

▪ **Dokumentation als wichtige Informationsquelle**

Die Dokumentation eines Anwendungssystems wird als wichtigste Informationsquelle zu dessen konzeptionellen und technischen Charakteristika angesehen. Sie umfasst schriftlich oder graphisch festgehaltene Informationen zu dem betreffenden Anwendungssystem.¹⁸⁸

Die Dokumentation eines Anwendungssystems kann nach den jeweiligen Hauptadressaten, den beschriebenen Aspekten und der Dokumentierungsmethode in verschiedene Teile untergliedert werden.¹⁸⁹ Aus Sicht der Anwendungsintegration erscheinen insbesondere drei Bereiche der Dokumentation zu den zu integrierenden Anwendungssystemen relevant: die Benutzerdokumentation, die Betreiberdokumentation und die Entwicklerdokumentation.¹⁹⁰

Wenngleich die Bedeutung der Dokumentation für den erfolgreichen Einsatz und die Wartung und Weiterentwicklung eines Anwendungssystems häufig betont wird und beispielsweise mit der DIN 66230 Normen zu ihrer Gestaltung existieren, variieren De-

ge oder keine Datenunabhängigkeit aus. Insbesondere zur Behebung dieser mangelnden Datenunabhängigkeit wurde seit den 70er Jahren zunehmend der Ansatz der Datenbanksysteme verfolgt. Dabei wurden zunächst insbesondere hierarchische Datenbanksysteme, später relationale und postrelationale Datenbanksysteme (z. B. objektorientierte oder multidimensionale Datenbanksysteme) eingesetzt (vgl. Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 215-222; Elmasri, Navathe /Database Systems/ 4-9).

¹⁸⁸ Vgl. Rupietta /Benutzerdokumentation/ 16. Der Begriff der Dokumentation wird sowohl für den Vorgang des Festhaltens der Informationen sowie für das aus diesem Vorgang resultierende Ergebnis verwendet (vgl. Mellis /Systemdokumentation/ 454). Da im Rahmen dieser Arbeit die Integration bestehender Anwendungssysteme betrachtet wird, wird Dokumentation lediglich als das Vorgangsergebnis verstanden.

¹⁸⁹ In der Literatur existieren diesbezüglich verschiedene und teilweise widersprüchliche Ansätze zur Abgrenzung und Bezeichnung einzelner Bereiche der Dokumentation. Vgl. beispielsweise Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 323f., Haupt /Dokumentation/ 252, Mellis /Systemdokumentation/ 454.

¹⁹⁰ Die Benutzerdokumentation richtet sich primär an die Benutzer eines Anwendungssystems und enthält Informationen zur Bedienung des Systems. Die Betreiberdokumentation richtet sich primär an die Betreiber eines Anwendungssystems und enthält Informationen zur Installation, Konfiguration und Administration der Anwendungssysteme (vgl. Mellis /Systemdokumentation/ 454; Stahlknecht, Hasenkamp /Wirtschaftsinformatik/ 324). Die Entwicklerdokumentation richtet sich primär an Systementwickler und enthält Informationen zu den extern wahrnehmbaren und den internen technischen Merkmalen der Anwendungssysteme. Diesbezüglich können beispielsweise die Dokumentation der Architektur eines Anwendungssystems aus verschiedenen Sichten aber auch die Kommentierung des Programmcodes genannt werden (vgl. Clements u. a. /Documenting Architectures/ 1-24; Balzert /Software-Technik/ 935-937).

tailgrad und Qualität der verfügbaren Dokumentation verschiedener Anwendungssysteme häufig stark von einander.¹⁹¹

2.1.6.2 Zentrale Herausforderungen für die Anwendungsintegration

Gemäß Hasselbring resultieren aus den Charakteristika der Anwendungssysteme drei zentrale Herausforderungen für die Anwendungsintegration: der Umgang mit Autonomie, Verteilung und Heterogenität.¹⁹²

Der Gegenstand der Anwendungsintegration ist die Verknüpfung von bestehenden Anwendungssystemen, die i. d. R. unabhängig voneinander entwickelt wurden und eigenständig im laufenden Geschäftsbetrieb der Unternehmen eingesetzt werden. Die Anwendungssysteme repräsentieren daher ein bestimmtes Verständnis ihrer Autonomie, das sowohl in ihren konzeptionellen und technischen Charakteristika als auch in ihrer Nutzung durch getrennte organisatorische Einheiten zum Ausdruck kommen kann. Mit der Feststellung des Integrationsproblems wird diese Autonomie hinterfragt, ihr wird ein Integrationsbedarf gegenüber gestellt. Gemäß Hasselbring resultieren daraus oftmals Konflikte, die nicht allein durch technische Mittel, sondern üblicherweise auch durch organisatorische Maßnahmen aufgelöst werden müssen.¹⁹³

Eine weitere zentrale Herausforderung für die Anwendungsintegration resultiert aus der Verteilung der Anwendungssysteme. Diese werden i. d. R. auf unterschiedlichen Rechnern und räumlich getrennt betrieben. Aus der Verteilung der Anwendungssysteme resultieren spezifische Ansprüche sowohl an die Integrationslösung als auch an die an ihrer Entwicklung beteiligten Personen.¹⁹⁴

Aus der Heterogenität, d. h. der Verschiedenartigkeit der konzeptionellen und technischen Charakteristika der Anwendungssysteme können wiederum Konflikte resultieren,

¹⁹¹ Vgl. Grupp /Dokumentieren/ 1-5. Die Qualität der Dokumentation basiert beispielsweise auf ihrer Verständlichkeit, Vollständigkeit, Zweckmäßigkeit, Fehlerfreiheit, Aktualität und maschinellen Weiterverarbeitbarkeit (vgl. Jung /Integrationsorientiertes Reengineering/ 21).

¹⁹² Vgl. Hasselbring /Integration/ 37.

¹⁹³ Vgl. Hasselbring /Integration/ 36f.

¹⁹⁴ Vgl. Hasselbring /Integration/ 36f.

die aufgelöst werden müssen.¹⁹⁵ Beispielsweise können verschiedene konzeptionelle Konflikte auftreten, wenn die Datenmodelle der zu integrierenden Anwendungssysteme voneinander abweichen obwohl sie die gleichen Realitätsausschnitte abbilden. Namenkonflikte resultieren beispielsweise aus Synonymen (d. h. unterschiedliche Bezeichnungen für Datenelemente, welche dieselben Informationen abbilden) oder Homonymen (d. h. gleichlautende Bezeichnungen für Datenelemente, welche unterschiedliche Informationen abbilden) in den Daten der Anwendungssysteme.¹⁹⁶ Konflikte zwischen den Spezifikationen der Datenelemente können ferner aus unterschiedlichen Definitionsbereichen und Formate der Daten resultieren oder hinsichtlich der Beziehungen zwischen den Datenelementen und der definierten Randbedingungen bestehen.¹⁹⁷ Die technische Heterogenität der Anwendungssysteme basiert beispielsweise auf der Verschiedenartigkeit der Systemsoftware, der systemnahen Software und der Hardwarekomponenten, die sie erfordern. Technische Konflikte können insbesondere aus unterschiedlichen Eigenschaften der externen Schnittstellen der Anwendungssysteme resultieren.¹⁹⁸

Im Zuge der Anwendungsintegration gilt es insbesondere, die Heterogenität der Anwendungssysteme zu kapseln und auf der Grundlage gemeinsamer Modelle, Strukturen oder Standards zu überbrücken.¹⁹⁹

2.2 Integrationslösungen

Durch die Anwendungsintegration soll eine Kongruenz von Integrationsbedarf und Integrationszustand erreicht werden - dem Integrationsproblem wird eine Integrationslösung entgegengestellt.

2.2.1 Integrationslösungen als einzigartige Softwaresysteme

Eine Integrationslösung wurde definiert als System, gebildet aus den verknüpften Anwendungssystemen und der Gesamtheit der Software, welche automatisierte Interaktionen der Anwendungssysteme unterstützt, sowie den zugehörigen Daten.

Eine Integrationslösung zeichnet sich damit durch die folgenden Charakteristika aus:

¹⁹⁵ Vgl. Hasselbring /Integration/ 37.

¹⁹⁶ Vgl. Morris u. a. /Database/ 298.

¹⁹⁷ Vgl. Morris u. a. /Database/ 298.

¹⁹⁸ Vgl. Hasselbring /Integration/ 37.

¹⁹⁹ Vgl. Hasselbring /Integration/ 36.

- Sie besteht aus mehreren Teilsystemen, die unabhängig voneinander betrieben werden. Insbesondere umfasst sie verschiedene Anwendungssysteme, die in der Regel weiterhin eigenständig zur Unterstützung bestimmter Anwendungsgebiete eingesetzt werden.
- Die einzelnen Teilsysteme einer Integrationslösung können geographisch stark verteilt sein.
- Die Integrationslösung weist funktionsbezogene und qualitätsbezogene Merkmale auf, die über diejenigen der einzelnen Teilsysteme hinausgehen.

Eine Integrationslösung kann somit als ein System von Softwaresystemen verstanden werden (engl. Systems of Systems, SoS).²⁰⁰ Ihr Aufbau wird als Integrationsarchitektur bezeichnet. In Anlehnung an den Begriff der Softwarearchitektur gemäß Bass, Clements und Kazman kann eine Integrationsarchitektur beschrieben werden anhand der Softwarekomponenten aus denen die Integrationslösung besteht, der nach Außen sichtbaren Eigenschaften dieser Komponenten und der Beziehungen zwischen ihnen.²⁰¹

Die Gestaltung einer Integrationslösung wird durch verschiedene Faktoren beeinflusst. Es liegt nahe, dass die Beschaffenheit des Integrationsproblems maßgeblich begründet, wie die Integrationslösung gestaltet wird. Indem mit der Integrationslösung definitionsgemäß das Ziel verfolgt wird, automatisierte Interaktionen zwischen Anwendungssystemen zu realisieren, können verschiedene grundlegende Gestaltungsmerkmale von Integrationslösungen bestimmt werden, deren Ausprägung sich i. d. R. an verschiedenen Gestaltungsmustern orientiert. Eine Integrationslösung wird ferner durch die eingesetzten Technologien geprägt, beispielsweise durch Auszeichnungssprachen, durch Standards und durch spezielle Standardsoftwaresysteme. All diese Aspekte werden in den anschließenden Unterkapiteln behandelt.

Auch wenn sie in der Fachliteratur zur Anwendungsintegration bislang kaum thematisiert werden, so ist zu vermuten, dass auch rechtliche Rahmenbedingungen (beispiels-

²⁰⁰ Vgl. Sage, Cuppan /Systems of Systems/ 326f. Sage und Cuppan nennen darüber hinaus als charakteristisches Merkmal von SoS, dass die Entwicklung des Gesamtsystems evolutionären Charakter hat. Es werden stets Teile hinzugefügt, verändert oder entfernt und damit Aufbau, Funktionalität und Zweck des SoS weiterentwickelt (vgl. Sage, Cuppan /Systems of Systems/ 326). Dieser Aspekt spiegelt sich auch in dem erläuterten zyklischen Charakter der Anwendungsintegration wieder. Eine Integrationslösung wird in dieser Arbeit jedoch im Sinne eines abgegrenzten Entwicklungsschrittes im Zuge der allgemeinen Integration der betrieblichen Anwendungssysteme verstanden.

²⁰¹ “[The] software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.” (Vgl. Bass, Clements, Kazman /Software Architecture/ 21)

weise handelsrechtliche Bestimmungen) und insbesondere die Interessen und Erfahrungen, der an der Gestaltung beteiligten Personen, die Gestaltung einer Integrationslösung prägen.²⁰² Aufgrund des komplexen Zusammenspiels all dieser Einflüsse steht am Ende der Anwendungsintegration stets eine in der Gesamtheit einzigartige Integrationslösung.²⁰³

2.2.2 Kategorien von Integrationslösungen

Auch wenn Integrationslösungen in ihrer Gesamtheit einzigartige Softwaresysteme darstellen, können sie anhand grundlegender Merkmale in verschiedene Kategorien eingeteilt werden. In der Fachliteratur werden dafür beispielsweise der angestrebte Integrationszustand, d. h. der Gegenstand der Integration, die Richtung und die Reichweite der Integration oder die Arten der zu integrierenden Anwendungssysteme herangezogen. Gemäß den jeweiligen Autoren sind die unterschiedenen Kategorien von Integrationslösungen mit charakteristischen Gestaltungsentscheidungen verbunden.

Linthicum misst dem Gegenstand der Integration maßgebliche Bedeutung für die Gestaltung einer Integrationslösung bei. Er unterscheidet diesbezüglich zwischen der informationsorientierten, der funktionsorientierten (bzw. serviceorientierten) und der geschäftsprozessorientierten Anwendungsintegration, die er mit charakteristischen Anforderungen an die Interaktionen der Anwendungssysteme und die Ausprägungen der funktionsbezogen Gestaltungsmerkmale einer Integrationslösung in Verbindung setzt.²⁰⁴

Die informationsorientierte Anwendungsintegration zielt darauf, Informationen zwischen den betroffenen Anwendungssystemen auszutauschen. Die serviceorientierte Anwendungsintegration hat zum Ziel, einzelne Funktionen der betroffenen Anwendungssysteme verfügbar zu machen (z. B. als Services) und sie flexibel zur neuartigen Unterstützung von Anwendungsgebieten kombinieren zu können. Die geschäftsprozessorientierte Anwendungsintegration schließlich bezieht sich darauf, mehrstufige automatisierte Interaktionsfolgen zwischen den betroffenen Anwendungssystemen zur

²⁰² Diese Annahmen basieren auf dem Modell des ‚architecture business cycle‘ gemäß Bass, Clements und Kazman (vgl. Bass, Clements, Kazman /Software Architecture/ 6-12) sowie dem Modell der Rahmenbedingungen der Softwareentwicklung gemäß Mellis (vgl. Mellis /Projektmanagement/ 41-43).

²⁰³ Vgl. Mische /Systems Integration/ 35.

²⁰⁴ Die folgenden Erläuterungen beziehen sich insbesondere auf Linthicum /Application Integration/ 6-19. Die Erläuterung der charakteristischen Anforderungen erfolgt im anschließenden Kapitel.

Abwicklung von Geschäftsprozessen zu realisieren. Dabei werden die in den einzelnen Anwendungssystemen abgebildeten Geschäftsprozessschritte erschlossen und kombiniert. Einer Integrationslösung können mehrere dieser Integrationsgegenstände zugrunde liegen.

Wangler und Paheerathan beziehen sich des Weiteren auf die Richtung und die Reichweite der Integration, zur Differenzierung verschiedener Kategorien von Integrationslösungen.²⁰⁵ Sie unterscheiden zwischen der horizontalen innerbetrieblichen Integration, der vertikalen innerbetrieblichen Integration und der zwischenbetrieblichen Integration. Die horizontale innerbetriebliche Integration zielt auf die Verknüpfung von Anwendungssystemen innerhalb eines Unternehmens, die verschiedene Stufen der betrieblichen Wertschöpfungskette unterstützen. Die vertikale innerbetriebliche Integration bezweckt dagegen die Verknüpfung von Anwendungssystemen, die verschiedenen organisatorischen Ebenen zuzuordnen sind. Die zwischenbetriebliche Integration schließlich betrifft die Verknüpfung von Anwendungssystemen verschiedener Unternehmen.

Die Autoren gelangen jedoch abschließend zum Erkenntnis, dass sich die unterschiedlichen Typen weniger stark in den an sie gestellten Anforderungen unterscheiden als zu erwarten wäre.

Die vertikale innerbetriebliche Integration zeichnet sich aus ihrer Sicht gegenüber der horizontalen innerbetrieblichen Integration insbesondere durch eine hohe Heterogenität der Anwendungssysteme aus, die beispielsweise in den unterschiedlichen Verdichtungsgraden der durch sie verarbeiteten Informationen zum Ausdruck kommen.²⁰⁶ Im Falle der zwischenbetrieblichen Integration sehen sie die Autonomie der Anwendungssysteme als zentrale Herausforderung ebenso wie höhere Anforderungen an die Sicherheit.

Themistocleous und Irani greifen ebenfalls die Reichweite der Integration als primäres Kriterium für die Entwicklung einer Taxonomy der Anwendungsintegration auf.²⁰⁷ Sie

²⁰⁵ Zum Ansatz von Wangler und Paheerathan vgl. im Weiteren Wangler, Paheerathan /Integration of Organizational IT Systems/ 82-89.

²⁰⁶ Während bei der horizontalen Anwendungsintegration der Informationsgehalt der Daten über die zusammengeführten Anwendungssysteme hinweg häufig gleich bleibt, muss er bei der vertikalen Integration oftmals im Zusammenhang mit steigenden organisatorischen Ebenen erhöht werden. Vgl. diesbezüglich auch Linß /Nutzeffekte/ 24; Mertens /Integrierte Informationsverarbeitung/ 4-6.

²⁰⁷ Vgl. im Weiteren Themistocleous, Irani /Taxonomy for Application Integration/.

unterscheiden auf oberer Ebene drei Kategorien von Integrationslösungen, die eine innerbetriebliche Anwendungsintegration, eine zwischenbetriebliche Anwendungsintegration oder eine hybride Anwendungsintegration bezwecken.²⁰⁸

Auf einer zweiten Ebene werden Integrationslösungen der innerbetrieblichen Anwendungsintegration danach unterschieden, ob sie die Anbindung von standardisierten Softwarepaketen (insbesondere ERP-Systeme) oder von individuellen Anwendungssystemen (insbesondere Altsysteme) betreffen. Diese beiden Typen basieren auf charakteristischen Integrationsproblemen (insbesondere bezüglich der konzeptionellen und technischen Charakteristika der Anwendungssysteme) und sind daher gemäß den Autoren mit dem Einsatz unterschiedlicher Technologien verbunden.

Die Integrationslösungen der hybriden Anwendungsintegration werden auf der zweiten Ebene lediglich als Business-to-Consumer (B2C) Systeme konkretisiert, bei denen überwiegend Integrationslösungen für internetbasierte Transaktionen bzw. Interaktionen mit Konsumenten bereitgestellt werden. Damit kann in unterschiedlichem Ausmaß eine innerbetriebliche oder zwischenbetriebliche Anwendungsintegration verbunden sein.

Der zwischenbetrieblichen Anwendungsintegration können gemäß den Autoren zum einen Integrationslösungen zugeordnet werden, die Geschäftsprozesse über die organisatorischen Grenzen hinweg unterstützen und somit erweiterte Unternehmen schaffen (engl. extended enterprises). Zum anderen werden hier Integrationslösungen eingeordnet, welche die informationstechnische Grundlage virtueller Unternehmen (engl. virtual enterprises) bilden. Diese virtuellen Unternehmen werden gegenüber den erweiterten Unternehmen mit einer engeren Integration, d. h. mit einer geringeren Autonomie der Anwendungssysteme gleichgesetzt.

2.2.3 Gestaltungsmerkmale einer Integrationslösung

Ausgehend vom Begriffsverständnis der Anwendungsintegration sollen Integrationslösungen automatisierte Interaktionen der beteiligten Anwendungssysteme zur systemübergreifenden Unterstützung von Geschäftsprozessen ermöglichen.

Als grundlegende Gestaltungsmerkmale einer Integrationslösung können diesbezüglich die Form der Interaktionen sowie verschiedene funktionsbezogene und qualitätsbezoge-

²⁰⁸ Diese Konzepte wurden bereits im Zuge der Erläuterung der Reichweite eines Integrationszustands vorgestellt.

ne Merkmale von Integrationslösungen unterschieden werden, auf denen diese Interaktionen basieren.



Abbildung 2-5: Gestaltungsmerkmale einer Integrationslösung

2.2.3.1 Automatisierte Interaktionen als Zweck von Integrationslösungen

Eine Interaktion beschreibt eine wechselseitige Handlung bzw. ein wechselseitiges Vorgehen von miteinander in Beziehung stehenden Akteuren.²⁰⁹ Im Kontext der Anwendungsintegration bedeutet dies, dass verschiedene Anwendungssysteme miteinander in Beziehung stehen und wechselseitig Aktionen ausführen. Diese wechselseitigen Aktionen basieren auf der Kommunikation zwischen den Softwarekomponenten einer Integrationslösung.²¹⁰

Es können unterschiedliche Formen von Interaktionen zwischen den Anwendungssystemen erfolgen. Um diese zu charakterisieren, können beispielsweise die Schicht der

²⁰⁹ Vgl. Wahrig Deutsche Rechtschreibung.

²¹⁰ Vgl. Mehta, Medvidovic, Phadke /Software Connectors/ 179; Vasconcelos u. a. /Architectural Framework/ 4.

Interaktion, die Beziehung zwischen den Anwendungssystemen, die Art der Kommunikation, das zugrunde liegende Kommunikationsmodell und der Interaktionsmechanismus betrachtet werden. Diese verschiedenen Betrachtungsperspektiven der Interaktionen stehen teilweise im Zusammenhang zueinander.

▪ **Interaktionsschicht**

Es wurde bereits erläutert, dass Anwendungssysteme idealtypisch in die drei Schichten der Benutzerschnittstelle, der Anwendungsfunktionalität und der Datenverwaltung untergliedert werden können. Eine Interaktion zwischen Anwendungssystemen kann nunmehr dahingehend charakterisiert werden, über welche dieser Schichten sie erfolgt. Entsprechend kann die Interaktion auf der Schicht der Benutzerschnittstelle, die Interaktion auf der Schicht der Anwendungsfunktionalität und die Interaktion auf der Schicht der Datenverwaltung unterschieden werden, wobei durchaus unterschiedliche Schichten der interagierenden Anwendungssysteme betroffen sein können.²¹¹

²¹¹ In der Fachliteratur existieren verschiedene Vorschläge zur Abgrenzung und Charakterisierung dieser Ansätze. Ferstl und Sinz beispielsweise unterscheiden zwischen der „datenflussorientierten Funktionsintegration“, der „Datenintegration“ und der „Objektintegration“ (vgl. Ferstl, Sinz /Wirtschaftsinformatik/ 220). Ruh, Maginnis und Brown differenzieren das „presentation integration model“, das „data integration model“ und das „functional integration model“ (vgl. Ruh, Maginnis, Brown /EAI/ 18-20). Keller nennt darüber hinaus noch die „Integration über Komponenten“ (vgl. Keller /EAI/ 68).

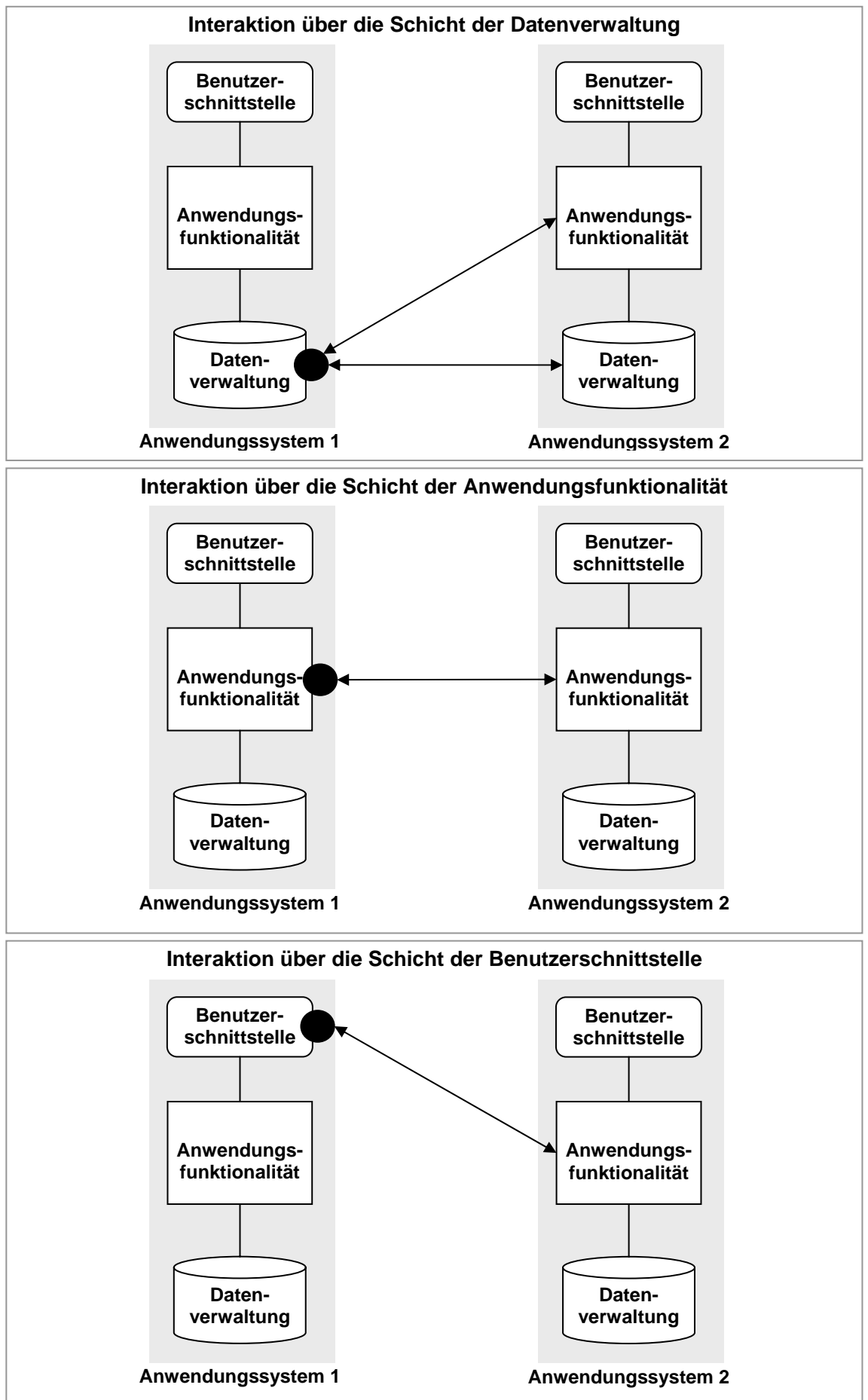


Abbildung 2-6: Alternative Interaktionsschichten

In Abhängigkeit von der betroffenen Schicht der Anwendungssysteme zeichnet sich eine Interaktion durch spezifische Merkmale aus.²¹² Insbesondere besteht ein Zusammenhang zwischen der Schicht der Interaktion und dem Integrationsgegenstand. Lediglich die Interaktion über die Schichten der Benutzerschnittstelle und der Anwendungsfunktionalität ermöglichen eine Integration von Funktionen, wobei über die Benutzerschnittstelle i. d. R. nur stark begrenzte Zugriffe auf die Funktionen eines Anwendungssystems möglich sind.²¹³ Die Integration auf der Schicht der Datenverwaltung ermöglicht zwar meist einen umfassenden Austausch von Daten, allerdings können die in der Funktionalität enthaltenen Zulässigkeits- und Plausibilitätsprüfungen nicht genutzt werden. Daraus resultiert ein Risiko für die Integrität der Daten.²¹⁴

Innerhalb einer Integrationsarchitektur kann für die Anbindung der verschiedenen Anwendungssysteme unterschiedliche Ansätze verfolgt werden, teilweise kann ein Anwendungssystem über mehrere Schichten mit anderen Anwendungssystemen interagieren.²¹⁵

▪ **Beziehung zwischen den Anwendungssystemen**

Es können unterschiedliche Beziehungen zwischen den Anwendungssystemen unterschieden werden. Im Falle einer Client/Server-Beziehung stellt ein Anwendungssystem als sog. Server Dienste bereit, die von mehreren anderen Anwendungssystemen, den sog. Clients, angefordert werden können. Eine Interaktion wird demnach stets von den Clients ausgelöst. Im Falle einer Peer-to-Peer-Beziehung können alle betroffenen Anwendungssysteme Interaktionen auslösen, d. h. als Client oder als Server agieren. Im Falle einer Push-Beziehung (bzw. Server/Client-Beziehung) löst ein Server die Interaktionen mit den Clients aus, in dem er selbständig Nachrichten an sie sendet (insbesondere im Sinne des sog. „publish and subscribe“ Modells).²¹⁶

²¹² Vgl. Ruh, Maginnis, Brown /EAI/ 18-20, Brodie, Stonebraker /Legacy Systems/ 21f., Riehm, Vogler /Middleware/ 29 sowie Keller /EAI/ 60.

²¹³ Vgl. Linthicum /EAI/ 80, Ruh, Maginnis, Brown /EAI/ 23.

²¹⁴ Vgl. Ruh, Maginnis, Brown /EAI/ 27.

²¹⁵ Vgl. Trapp, Otto /EAI/ 109; Linthicum /Application Integration/ 32.

²¹⁶ Vgl. Britton /IT-Architecture/ 81-82.

Ferner kann eine unmittelbare Beziehung zwischen den Anwendungssystemen bestehen, oder die Interaktion erfolgt mittelbar über ein zwischengeschaltetes Softwaresystem, d. h. über einen Intermediär.²¹⁷

▪ **Zweck der Kommunikation**

Die Kommunikation zwischen den Softwarekomponenten einer Integrationslösung kann unterschiedliche Zwecke verfolgen. In Anlehnung an die Theorie der Sprechakte nach Searle²¹⁸ werden in der von dem Unternehmen Viewlocity entwickelten Business Model Language (BML) beispielhaft verschiedene Kategorien von Kommunikationsakten unterschieden, die häufig innerhalb von Integrationslösungen auftreten.²¹⁹ Demnach kann eine Softwarekomponente eine Aufforderung an eine andere Softwarekomponente senden. Eine solche Aufforderung kann sich auf Informationen, Dienste sowie die Reservierung oder Inanspruchnahme von Ressourcen beziehen. Als Reaktion auf eine erhaltene Aufforderung kann eine Antwort versendet werden. In dieser können entweder die angeforderten Informationen übermittelt oder das Erbringen eines Dienstes bzw. die Reservierung oder Zuordnung von Ressourcen bestätigt werden. Eine Softwarekomponente kann zudem das zukünftige Erbringen eines Dienstes als Reaktion auf eine entsprechende Anfrage zusagen. Des Weiteren kann eine Softwarekomponente eine Mitteilung verschicken, mittels derer ein Empfänger über eine bestimmte Zustandsänderung informiert wird. Schließlich kann die Aufhebung einer Reservierung oder Zuordnung von Ressourcen übermittelt werden.

▪ **Zugrunde liegendes Kommunikationsmodell**

Die Kommunikation der Anwendungssysteme kann auf verschiedenen Kommunikationsmodellen basieren, die teilweise im Zusammenhang mit den unterschiedenen Beziehungen zwischen den Anwendungssystemen stehen. So kann sie beispielsweise synchron oder asynchron stattfinden. Ein synchrones Kommunikationsmodell geht von zeitlich gleichlaufenden Aktionen der Softwarekomponenten im Rahmen ihrer Kommunikation aus, d. h. die Komponenten sind im Ablauf der Kommunikation zeitlich aneinander gekoppelt. Ein asynchrones Kommunikationsmodell geht davon aus, dass

²¹⁷ Vgl. Andersson, Johnson /Integration Styles/ 230f.

²¹⁸ Vgl. Searle /Speech Acts/ 22-53.

²¹⁹ Zur Abgrenzung und Charakterisierung dieser Kategorien vgl. Johannesson, Perjons /Process Modeling/ 172-174.

die Aktionen der Softwarekomponenten im Rahmen der Kommunikation zeitlich nicht gleichlaufend sind.²²⁰

Die Kommunikation kann ferner verbindungsorientiert oder verbindungslos erfolgen. Bei einer verbindungsorientierten Kommunikation identifiziert Softwarekomponente A Softwarekomponente B, bindet sich an sie an und teilt so lange einen gemeinsamen Zustand mit ihr, bis der Kommunikationsvorgang abgeschlossen ist. Bei einer verbindungslosen Kommunikation hingegen ist es nicht erforderlich, dass Softwarekomponente A Softwarekomponente B identifiziert und sich an sie bindet. I. d. R. weiß Softwarekomponente A nicht, ob die Nachricht überhaupt von einer anderen Softwarekomponente empfangen und verarbeitet wird oder wann dies geschehen wird.²²¹

Schließlich kann sich die Kommunikation zwischen zwei („one-to-one“) oder vielen Anwendungssystemen („many-to-one“, „one-to-many“ oder „many-to-many“) ereignen.²²²

▪ **Interaktionsmechanismus**

Die Interaktion zwischen den Anwendungssystemen kann aufgrund verschiedener Interaktionsmechanismen erfolgen. Diesbezüglich können beispielsweise die Interaktion mittels Dateien, Nachrichten, Prozeduraufrufen oder verteilten Objekten unterschieden werden.²²³ Diese verschiedenen Interaktionsmechanismen stehen in Zusammenhang mit den erläuterten Kommunikationsmodellen. Die Interaktion mittels Dateien basiert auf den Fähigkeiten der teilnehmenden Anwendungssysteme, Dateien oder andere große Datenblöcke zu erstellen und zu verarbeiten. Die Interaktion mittels Nachrichten bedeutet, dass der Austausch von Daten oder Kontrolle in Form kleiner Informationseinheiten mit einheitlicher Struktur erfolgt. Dabei können die Nachrichten entweder direkt übertragen werden oder sie werden in eine Warteschlange eingestellt, um von dort abgerufen zu werden. Sowohl die Interaktion mittels Dateien als auch diejenige mittels Nachrichten basiert auf einem asynchronen Kommunikationsmodell.²²⁴ Demgegenüber basiert die Interaktion mittels Prozeduraufrufen auf einem synchronen Kommunikationsmodell. Bei diesem Interaktionsmechanismus erhalten die Anwendungssysteme die Möglich-

²²⁰ Vgl. Keller /EAI/ 76f.

²²¹ Vgl. Keller /EAI/ 78f.

²²² Vgl. Linthicum /Application Integration/ 117-120; Lutz /patterns/ 67.

²²³ Vgl. Ließmann /Schnittstellenorientierung/ 2; Goodyear /Enterprise System Architecture/ Kapitel 3; S. 27.

²²⁴ Vgl. Linthicum /Application Integration/ 126-128; Britton /IT-Architecture/ 34-36.

keit, Funktionen anderer Anwendungssysteme aufzurufen, als handle es sich um lokale Funktionen.²²⁵ Die Interaktion mittels verteilter Objekte basiert auf dem objektorientierten Entwicklungsparadigma. Sie ähnelt der Interaktion mittels Prozeduraufrufen, allerdings erhalten die Anwendungssysteme die Möglichkeit, die Methoden von Objekten in anderen Anwendungssystemen aufzurufen.²²⁶

2.2.3.2 Funktionsbezogene Gestaltungsmerkmale

Die Fähigkeit, automatisierte Interaktionen zwischen den beteiligten Anwendungssystemen zur systemübergreifenden Unterstützung von Geschäftsprozessen zu ermöglichen, resultiert aus verschiedenen funktionsbezogenen Gestaltungsmerkmalen von Integrationslösungen.

In der Fachliteratur existieren verschiedene Ansätze zur Differenzierung dieser Merkmale. Linthicum beispielsweise unterscheidet die Verbindungsfähigkeit, die Transformation, die Weiterleitung von Informationen, die Speicherung von Informationen sowie die Unterstützung von Transaktionen und Geschäftsprozessen als erforderliche funktionsbezogene Merkmale einer Integrationslösung. Er bezeichnet sie zusammenfassend als „Manifest der Anwendungsintegration“.²²⁷ Trapp und Otto unterscheiden verschiedene Schichten einer Integrationsarchitektur, denen sie verschiedene funktionsbezogene Merkmale zuordnen. Im Einzelnen sind dies die Kommunikationsschicht, die Transformations- und Formatierungsschicht, die Applikationsanbindungsschicht und die Geschäftsprozess-Management-Schicht. Dabei berufen sie sich auf eine nicht näher erläuterte Studie des Unternehmens Accenture, gemäß der die genannten Merkmale grundlegend für jegliche Integrationslösungen seien.²²⁸ Keller unterscheidet in ähnlicher Weise eine Protokoll-Adapterschicht, eine Kommunikationsschicht und eine Prozessschicht von Integrationsarchitekturen.²²⁹

Aufbauend auf diesen Ansätzen werden in dieser Arbeit die Anbindung, die Übermittlung von Informationen, die Transformation und Anreicherung von Informationen, die Interaktionsauslösung und -steuerung sowie der Benutzerdialog als funktionsbezogene

²²⁵ Vgl. Linthicum /Application Integration/ 125; Britton /IT-Architecture/ 25-27.

²²⁶ Vgl. Linthicum /Application Integration/ 128; Britton /IT-Architecture/ 49-52.

²²⁷ Die Bezeichnungen im englischsprachigen Original lauten connectivity, transformation, information routing, persistence, transactionality und process awareness (vgl. Linthicum /Application Integration/ 405-422).

²²⁸ Vgl. Trapp, Otto /EAI/ 109f.

²²⁹ Vgl. Keller /EAI/ 43f.

Gestaltungsmerkmale von Integrationslösungen unterschieden. Die Festlegung ihrer Ausprägungen und insbesondere ihre Aufteilung auf die Komponenten einer Integrationslösung (d. h. inwiefern sie durch die Anwendungssysteme selbst oder durch über- bzw. zwischengeordnete Softwarekomponenten realisiert werden) stellen zentrale Gestaltungsentscheidungen dar. Jung spricht in diesem Zusammenhang von den „Parametern einer Verteilungsarchitektur“.²³⁰

▪ **Anbindung der Anwendungssysteme**

Durch die Anbindung werden die grundlegenden technischen Voraussetzungen dafür geschaffen, dass ein Anwendungssystem mit anderen Anwendungssystemen interagieren kann. Bestehende externe Schnittstellen der Anwendungssysteme werden erschlossen oder neue externe Schnittstellen geschaffen. Damit wird der Austausch von Daten und Kontrolle, d. h. das Versenden und Empfangen von Aufforderungen, Antworten, Mitteilungen oder Aufhebungs-Nachrichten ermöglicht.

▪ **Übermittlung von Informationen**

Im Zuge ihrer Interaktion kommunizieren die Anwendungssysteme miteinander, d. h. sie tauschen Informationen aus. Zunächst müssen die erforderlichen Interaktionspartner identifiziert, lokalisiert und die Eigenschaften ihrer Anbindungspunkte bestimmt werden.²³¹ Darauf aufbauend muss die Verfügbarkeit der Interaktionspartner geprüft werden.²³² Die Kommunikation der Anwendungssysteme kann auf verschiedenen der bereits erläuterten Kommunikationsmodelle basieren. Oftmals erfordert die Interaktion der Anwendungssysteme zudem, dass bei der Übermittlung der Informationen das Transaktions-Prinzip unterstützt wird.²³³ Bei der Übermittlung von Informationen sind auch

²³⁰ Jung /Integrationsorientiertes Reengineering/ 6.

²³¹ Erfolgt die Identifikation der Interaktionspartner allein aufgrund der Analyse der zu übermittelnden Informationen anhand definierter Regelungen, so wird dies im Allgemeinen als „intelligent routing“ bezeichnet (vgl. Linthicum /Application Integration/ 416).

²³² Vgl. Andersson, Johnson /Integration Styles/ 227; Linthicum /Application Integration/ 416; Oey u. a. /SOA/ 157.

²³³ Vgl. Linthicum /Application Integration/ 418-420. Eine ordnungsgemäße Transaktion zeichnet sich durch die sogenannten „ACID-Eigenschaften“ aus. Ununterbrechbarkeit (engl. atomicity) bedeutet, dass eine Transaktion atomar ist, d. h. eine Transaktion ist unteilbar – entweder sie wird vollständig abgewickelt oder gar nicht. Konsistenz (engl. consistency) bedeutet, dass sichergestellt wird, dass sich die beteiligten Systeme immer in einem konsistenten Zustand befinden. Isolierte Zurücksetzbarkeit (engl. isolation) bedeutet, dass die in einer laufenden Transaktion vorgenommenen Änderungen an den Systemzuständen so lange als vorläufig gelten, wie die Transaktion noch nicht vollständig und erfolgreich abgeschlossen ist. Bis zu diesem Zeitpunkt können sie unabhängig von anderen Transaktionen wieder vollständig rückgängig gemacht werden. Dauerhaftigkeit (engl. durability) schließlich

Sicherheitsaspekte von Bedeutung, das heißt es muss ein konsistentes Sicherheitskonzept unterstützt werden.²³⁴ Durch eine Protokollierung der Kommunikation und die persistente Speicherung ausgetauschter Informationen kann die Interaktion der Anwendungssysteme zudem überwacht und nachvollzogen werden.²³⁵

▪ **Transformation und Anreicherung von Informationen**

Es kann erforderlich sein, die zwischen den Anwendungssystemen ausgetauschten Informationen anhand definierter Regelungen inhaltlich zu verändern, bevor sie übermittelt werden. Zur Auflösung konzeptioneller Konflikte zwischen den Anwendungssystemen erfolgt eine Transformation der Informationen im Hinblick auf Syntax oder Semantik. Dabei können Informationen verschiedenen Ursprungs zusammengefügt oder bestimmte Informationen ausgefiltert werden. Darüber hinaus können Informationen aufgrund von Fachfunktionalität vorverarbeitet und damit angereichert werden.²³⁶

▪ **Interaktionsauslösung und -steuerung**

Die Interaktion von Anwendungssystemen muss im rechten Moment, d. h. beim Eintreten definierter Ereignisse, ausgelöst werden. Ereignisse können beispielsweise das Aufrufen einer Funktion, die Modifikation von Datenbeständen, zeitliche Ereignisse oder Kombinationen verschiedener dieser Ereignistypen sein.²³⁷ Ggf. wird die Auslösung einer Interaktion beim Eintreten eines Ereignisses mit bestimmten Bedingungen verbunden.²³⁸

Darüber hinaus kann es erforderlich sein, den Ablauf der Interaktionen der Anwendungssysteme zu koordinieren und zu überwachen. Dies ist insbesondere dann von Bedeutung, wenn eine automatisierte Abwicklung mehrstufiger Interaktionsfolgen angestrebt wird, d. h. wenn der Integrationsgegenstand des angestrebten Integrationszustands auch die Prozessabwicklung umfasst.

bedeutet, dass die Systemzustände nach der vollständigen Abwicklung einer Transaktion dauerhaft sind und auch Systemfehler überstehen (vgl. Linthicum /Application Integration/ 419; Keller /EAI/ 97f.).

²³⁴ Vgl. Trapp, Otto /EAI/ 108 sowie die anschließende Erläuterung der qualitätsbezogenen Gestaltungsmerkmale von Integrationszuständen.

²³⁵ Vgl. Linthicum /Application Integration/ 416-418.

²³⁶ Vgl. Trapp, Otto /EAI/ 109; Linthicum /Application Integration/ 412-416.

²³⁷ Vgl. Dittrich, Gatzju, Geppert /ADBMS/ 42.

²³⁸ Bedingungen beschreiben die Zustände, in denen sich die Anwendungssysteme oder sonstige Komponenten einer Integrationsarchitektur befinden müssen, damit eine bestimmte Aktion beim Eintreten definierter Ereignisse ausgelöst wird (in Anlehnung an Dittrich, Gatzju, Geppert /ADBMS/ 43).

Die Steuerung der Interaktion erfordert, dass auf Ereignisse reagiert wird und neue Ereignisse generiert werden können. Dabei muss angemessen auf Ausnahmen und Fehler reagiert werden können.²³⁹ Auch das sog. „Load-Balancing“, im Sinne einer automatischen Verteilung der Verarbeitungslast auf verschiedene Systeme bei Lastspitzen kann dabei erforderlich sein.²⁴⁰ Verbunden mit der Interaktionssteuerung kann ferner die Anzeige der aktuellen Statusinformationen zu den Interaktionen (sog. „Monitoring“) oder die Sammlung und Aufbereitung statistischer Daten (sog. „Reporting“), wie z. B. Durchsatzraten, Fehlerfällen, Zeitverbrauch unterstützt werden.²⁴¹

▪ **Benutzerdialog**

Im Zuge der Abwicklung der unterstützten Geschäftsprozesse ist häufig neben der automatisierten Interaktion der Anwendungssysteme auch ein Dialog mit Benutzern erforderlich oder im Zuge des Monitoring bzw. Reporting erwünscht. Sind die Benutzerschnittstellen der Anwendungssysteme dafür nicht ausreichend oder geeignet, müssen sie ergänzt oder alternative Benutzerschnittstellen geschaffen werden. Darüber hinaus können Möglichkeiten zur Konfiguration, zur Administration, zur Überwachung und zum Fehlermanagement der Komponenten der Integrationsarchitektur angeboten werden.²⁴²

2.2.3.3 Qualitätsbezogene Gestaltungsmerkmale

Neben den funktionsbezogenen Merkmalen zeichnet sich eine Integrationslösung durch bestimmte Ausprägungen qualitätsbezogener Merkmale aus. Diese sind mit den Ausprägungen der Qualitätsmerkmale des durch die Integrationslösung repräsentierten Integrationszustands gleichzusetzen.

Die Zuverlässigkeit des durch die Anwendungsintegration erreichten Integrationszustands wird beispielsweise durch die Konsistenz, die Verfügbarkeit und die Wiederherstellbarkeit der Integrationslösung beeinflusst.²⁴³ Das Merkmal der Konsistenz beschreibt, inwiefern die einzelnen Anwendungssysteme, d. h. die einzelnen Software-

²³⁹ Vgl. Linthicum /Application Integration/ 64.

²⁴⁰ Vgl. Buhl, Christ, Pape /Marktstudie/ 13.

²⁴¹ Vgl. Buhl, Christ, Pape /Marktstudie/ 13.

²⁴² Vgl. Kaib /EAI/ 167.

²⁴³ Vgl. DIN /66272/ 8.

komponenten und Daten, gemeinsam ein einheitliches und zulässiges Abbild der betrieblichen Realität darstellen.²⁴⁴ Verfügbarkeit (engl. availability) beschreibt das Verhältnis der Zeit, in der die Integrationslösung wie beabsichtigt die Interaktion der Anwendungssysteme unterstützt, zu der Zeit, in der sie aufgrund von Fehlern oder Fehlerbehebungen gestört ist oder beispielsweise aufgrund von Datensicherungs-, Administrations-, Konfigurations- und Wartungstätigkeiten bewusst aus dem Betrieb genommen wird. Die Verfügbarkeit einer Integrationslösung nimmt also mit zunehmenden Ausfall- und Stillstandszeiten ab.²⁴⁵ Die Gewährleistung der Wiederherstellbarkeit (eng. recoverability) einer Integrationslösung bedeutet, dass das ursprüngliche Leistungsniveau der Integrationslösung nach Eintreten eines Fehlers wieder hergestellt und auch die direkt betroffenen Daten wieder gewonnen werden können.²⁴⁶ Schließlich kann der Zuverlässigkeit kann auch zugeordnet werden, inwiefern die Integrationslösung geeignet ist, das anfallende Interaktionsvolumen zu bewältigen. Diesbezüglich ist beispielsweise der Umfang der zwischen den Anwendungssystemen zu übertragenden Daten oder die Anzahl parallel auftretender Interaktionen von Bedeutung.²⁴⁷ Die Performanz der Integrationslösung beschreibt damit die Geschwindigkeit, mit der Interaktionen zwischen Anwendungssystemen abgewickelt werden.

Das Zeitverhalten des durch die Anwendungsintegration erreichten Integrationszustands wird beispielsweise durch die Performanz der Softwarekomponenten der Integrationslösung beeinflusst. Die Performanz beschreibt die Antwortzeit, welche ein Softwaresystem benötigt, um auf einen Stimulus zu reagieren oder die Anzahl von Aktivitäten, welche ein Softwaresystem in einem bestimmten Zeitintervall durchführen kann.²⁴⁸

Die Flexibilität des durch die Anwendungsintegration erreichten Integrationszustands wird beispielsweise durch die Integrierbarkeit, die Skalierbarkeit und die Aufrüstbarkeit

²⁴⁴ Vgl. Ferstl, Sinz /Wirtschaftsinformatik/ 219. Es können zwei Formen der Konsistenz unterschieden werden. Die semantische Konsistenz beschreibt, inwiefern die in den Daten der Anwendungssysteme abgebildeten Informationen zur betrieblichen Realität einheitlich und zulässig sind. Die semantische Konsistenz steht somit auch in Zusammenhang mit der Aktualität der in den Anwendungssystemen verarbeiteten Daten. Die operationale Konsistenz beschreibt, inwiefern die Anwendungssysteme vor und nach einer Interaktion ein einheitliches und zulässiges Abbild der betrieblichen Realität darstellen (vgl. Ferstl, Sinz /Wirtschaftsinformatik/ 219).

²⁴⁵ Vgl. Bass, Clements, Kazman /Software Architecture/ 80; Britton /IT-Architecture/ 120f.

²⁴⁶ Vgl. Cummins /Enterprise Integration/ 320; DIN /66272/ 8.

²⁴⁷ Vgl. Lam, Shankararaman /Enterprise Integration/ 44.

²⁴⁸ Vgl. Bass, Clements, Kazman /Software Architecture/ 79; Lam, Shankararaman /Enterprise Integration/ 44.

der Integrationslösung beeinflusst. Integrierbarkeit (engl. integrability) beschreibt die Möglichkeit, mit geringem Aufwand neue Anwendungssysteme oder Integrationssysteme in die Integrationsarchitektur aufzunehmen.²⁴⁹ Skalierbarkeit (engl. scalability) beschreibt die Fähigkeit einer Integrationsarchitektur, Änderungen zu unterstützen, die eine starke Vergrößerung der Integrationslösung bedeuten, entweder hinsichtlich der Anzahl der verbundenen Anwendungssysteme oder hinsichtlich des Ausmaßes oder der Form der Interaktion zwischen den Anwendungssystemen.²⁵⁰ Aufrüstbarkeit (engl. upgradeability) beschreibt die Fähigkeit einer Integrationsarchitektur, die Aufrüstung einzelner Komponenten, im Sinne des Anpassens an einen fortgeschrittenen Entwicklungsstand, zu unterstützen.²⁵¹ Die Flexibilität wird ferner dadurch beeinflusst, wie eng die integrierten Anwendungssysteme miteinander gekoppelt sind.²⁵² Je enger die Kopplung der Softwarekomponenten einer Integrationslösung, desto größer die Auswirkungen, welche Änderungen an einer Komponente auf die anderen Komponenten haben.²⁵³

Die Sicherheit des durch die Anwendungsintegration erreichten Integrationszustands wird beispielsweise durch die Unterstützung der Authentifizierung, der Autorisierung, der Kommunikationssicherheit, der Gewährleistung der Integrität der Anwendungssysteme sowie der Unleugbarkeit und Nachvollziehbarkeit charakterisiert. Authentifizierung bedeutet, dass die Benutzer sicherheitsrelevanter Aspekte der Systeme zuverlässig identifiziert werden können.²⁵⁴ Autorisierung erfordert, dass die Nutzungsrechte der Benutzer spezifisch festgelegt werden können und die Einhaltung dieser Regelungen sichergestellt wird. Der Aspekt der Autorisierung baut damit auf dem Aspekt der Authentifizierung auf.²⁵⁵ Kommunikationssicherheit betrifft beispielsweise die Gewährleistung der Abhörsicherheit/Abfangsicherheit der elektronischen Nachrichten sowie die Gewährleistung, dass bestimmte Nachrichten zuverlässig übermittelt werden und der

²⁴⁹ Vgl. Kazman, Bass /Quality Attributes/ 29; Lam, Shankararaman /Enterprise Integration/ 46.

²⁵⁰ Vgl. Kazman, Bass /Quality Attributes/ 28.

²⁵¹ Vgl. Andersson, Johnson /Integration Styles/ 227.

²⁵² Die Kopplung wird im Kontext der Softwaresysteme aus zwei Perspektiven betrachtet. Zum einen im Hinblick auf die Stärke der Bindung der einzelnen internen Elemente einer Komponente, auch als Kohäsion bezeichnet. Zum anderen im Hinblick auf die Bindung zwischen den verschiedenen Komponenten, auch als Adhäsion bezeichnet (vgl. Heylinger /Coupling/ 220). Im Rahmen dieser Arbeit ist insbesondere die Kopplung im Sinne der Adhäsion relevant. Das im Rahmen dieser Arbeit vertretene Begriffsverständnis der Kopplung weicht somit von dem Linthicums ab (vgl. Linthicum /Application Integration/ 27f.).

²⁵³ Vgl. Ruh, Maginnis, Brown /EAI/ 21.

²⁵⁴ Vgl. Cummins /Enterprise Integration/ 321f.

²⁵⁵ Vgl. Cummins /Enterprise Integration/ 322f.

Absender der Nachrichten zuverlässig festgestellt werden kann.²⁵⁶ Die Gewährleistung der Integrität der Anwendungssysteme bedeutet, dass sichergestellt wird, dass die Systeme nur zulässige Operationen ausführen und dass keine Möglichkeiten zum Ausführen unzulässiger Operationen geboten werden.²⁵⁷ Unleugbarkeit (engl. non-repudiation) und Nachvollziehbarkeit ist insbesondere von Bedeutung, wenn durch die Anwendungsintegration die geschäftliche Interaktion zwischen verschiedenen organisatorischen Einheiten unterstützt werden soll. In diesem Fall ist es erforderlich, dass vertraglich relevante Sachverhalte den Parteien zuverlässig zugeordnet und nachvollzogen werden können.²⁵⁸

2.2.4 Integrationsarchitekturen

2.2.4.1 Gestaltungsmuster und Middleware

Die Entwicklung einer Integrationslösung erfordert die Ausgestaltung der zuvor erläuterten Merkmale und insbesondere deren Aufteilung auf die Komponenten der Integrationsarchitektur. Dies führt zu ähnlichen Problemen, zu deren Lösung oftmals einheitliche Lösungsschemata verwendet werden. Trotz der großen Vielfalt von Integrationslösungen können daher gemeinsame Gestaltungsmuster identifiziert werden (auch als Integrationsmuster bezeichnet).²⁵⁹

Gemäß Buschmann u. a. beschreibt ein Muster für Architekturen von Softwaresystemen „ein bestimmtes wiederkehrendes Gestaltungsproblem, das in spezifischen Gestaltungskontexten auftritt, und zeigt ein bewährtes generisches Schema zu dessen Lösung auf. Dieses Lösungsschema wird spezifiziert durch die zugrunde liegenden Komponenten, ihre Aufgaben und Beziehungen und die Formen ihres Zusammenwirkens.“²⁶⁰

Einzelne Muster können verschiedene Varianten aufweisen. Diese konkretisieren das Muster hinsichtlich des Gestaltungsproblems oder des Lösungsschemas, stellen aber weiterhin generische Baueinheiten zur Gestaltung von Integrationsarchitekturen dar.

²⁵⁶ Vgl. Cummins /Enterprise Integration/ 323.

²⁵⁷ Vgl. Cummins /Enterprise Integration/ 319f.

²⁵⁸ Vgl. Cummins /Enterprise Integration/ 324.

²⁵⁹ Vgl. beispielsweise Klesse, Wortman, Schelp /Applikationsintegration/ 262.

²⁶⁰ “A pattern for software architecture describes a particular recurring design problem that arises in specific design contexts, and presents a well-proven generic scheme for its solution. The solution scheme is specified by describing its constituent components, their responsibilities and relationships, and the ways in which they collaborate.” (Buschmann u. a. /Patterns/ 8).

Ferner können Muster oftmals selbst durch mehrere Muster niedriger Abstraktionsebenen beschrieben, d. h. verfeinert werden. Zudem können aus der Anwendung eines Musters zur Lösung eines Gestaltungsproblems neue Gestaltungsprobleme resultieren, zu deren Lösung wiederum Muster angewandt werden können.²⁶¹

Gegenwärtig existieren in der Fachliteratur nur wenige Ansätze zur gezielten Definition von Gestaltungsmustern für Integrationsarchitekturen. Diesbezüglich können beispielsweise die Ansätze von Lutz, von Andersson und Johnson sowie von Losavio, Ortega und Pérez angeführt werden.²⁶² Häufiger werden in der Fachliteratur verschiedene Kategorien von Softwarekomponenten unterschieden, die im Rahmen einer Integrationsarchitektur eine Interaktion zwischen Anwendungssystemen unterstützen.²⁶³ Da diese Kategorien von Softwarekomponenten auf bewährten Lösungsschemata für wiederkehrende Gestaltungsprobleme basieren, können sie ebenfalls als Repräsentanten bestimmter Gestaltungsmuster für Integrationsarchitekturen verstanden werden. In Tabelle 2-4 werden einige Gestaltungsmuster für Integrationsarchitekturen dargestellt.

²⁶¹ Vgl. Buschmann u. a. /Patterns/ 16-18; Andersson, Johnson /Integration Styles/ 227; Lutz /Patterns/ 66.

²⁶² Vgl. Lutz /Patterns/; Andersson, Johnson /Integration Styles/; Losavio, Ortega, Pérez /EAI/. Dabei werden jedoch weder die zu lösenden Gestaltungsprobleme noch die bewährten Lösungsschemata in formaler Weise beschrieben.

²⁶³ Vgl. beispielsweise Kaib /EAI/ 100-118, 135-151; Lublinsky /EAI Implementation/ 30f. Dabei werden die Kategorien der Softwarekomponenten unterschiedlich abgegrenzt und charakterisiert. Zudem werden teilweise unterschiedliche Begriffe verwendet.

Muster	Gestaltungsproblem	Lösungsschema
Adapter ²⁶⁴	Die externen Schnittstellen der Anwendungssysteme weisen spezifische Eigenschaften auf. Diese Heterogenität muss bei der Anbindung der Anwendungssysteme überbrückt werden.	<p>Der externen Schnittstelle eines Anwendungssystems wird eine Softwareschicht vorgeschaltet, um deren Kommunikationsregeln in eine andere Ausprägung zu konvertieren, ohne dass die externe Schnittstelle selbst geändert werden muss.</p> <p>Es können verschiedene Varianten dieses Musters unterschieden werden, die sich z. B. im Hinblick auf den funktionalen Umfang des Adapters (sog. „thin adapter“ oder „thick adapter“), die Konfiguration des Adapters (sog. „static adapter“ oder „dynamic adapter“) oder die Lokalisierung des Adapters (sog. „centralized adapter“ oder „distributed adapter“).</p>
Screen Scraper ²⁶⁵	Ein zu integrierendes Anwendungssystem besitzt keine zur Anbindung geeignete externe Schnittstelle. Dieses Problem kann insbesondere im Zusammenhang mit Altanwendungen auftreten, die zudem typischerweise textbasierte Benutzerschnittstellen aufweisen.	Eine vorgeschaltete Softwareschicht simuliert gegenüber dem zu integrierenden Anwendungssystem einen Benutzerdialog, wobei die Datenströme der Benutzerschnittstelle abgefangen und verarbeitet werden. Gegenüber den Softwaresystemen, mit denen das Anwendungssystem interagieren soll, schafft sie eine externe Schnittstelle.
Integration Messenger ²⁶⁶	Zwischen den zu integrierenden Anwendungssystemen sollen Informationen übermittelt oder Funktionsaufrufe vermittelt werden. Die dabei bestehenden Anforderungen an die Kommunikation werden durch die Anwendungssysteme nicht oder nicht einheitlich unterstützt.	<p>Eine zwischengeschaltete Softwareschicht realisiert die Kommunikation zwischen den Anwendungssystemen in der erforderlichen Weise.</p> <p>Es können verschiedene Varianten dieses Musters unterschieden werden, die sich beispielsweise im Hinblick auf die durch den Integration Messenger unterstützten Beziehungsformen, Interaktionsmechanismen und Kommunikationsmodelle unterscheiden.</p>

²⁶⁴ Vgl. Lutz /Patterns/ 67; Andersson, Johnson /Integration Styles/ 229; Linthicum /Application Integration/ 217-232; Rine, Nada, Jaber /Adapters/ 38-40.

²⁶⁵ Vgl. Ruh, Maginnis, Brown /EAI/ 19-22; Linthicum /EAI/ 83-85.

²⁶⁶ Vgl. Lutz /Patterns/ 67-70; Linthicum /Application Integration/ 115-134.

Muster	Gestaltungsproblem	Lösungsschema
Integration Server ²⁶⁷	Zwischen den zu integrierenden Anwendungssystemen sollen Informationen übermittelt oder Funktionsaufrufe vermittelt werden, wobei die ausgetauschten Daten auch transformiert und angereichert werden müssen. Auch die Interaktionsauslösung und -steuerung wird nicht in erforderlicher Weise unterstützt.	<p>Die Anwendungssysteme interagieren indirekt über ein zentrales Softwaresystem, das die erforderlichen funktionsbezogenen Merkmale in sich vereint.</p> <p>Es können verschiedene Varianten dieses Musters unterschieden werden, die sich beispielsweise im Hinblick auf die unterstützten Beziehungsformen, Interaktionsmechanismen und Kommunikationsmodelle unterscheiden. Weitere Varianten unterscheiden sich darin, ob der Integration Server die einzelnen Interaktionen zwischen den Anwendungssystemen im Kontext zueinander (sog. „stateful integration“) oder unabhängig voneinander (sog. „stateless integration“) behandelt.</p>
Process Automator ²⁶⁸	Im Zuge einer prozessorientierten Anwendungsintegration soll die durchgängige Unterstützung von mehrstufigen Geschäftsprozessen oder Geschäftsprozessfolgen unterstützt werden.	Die Auslösung und Steuerung der Interaktionen zwischen den Anwendungssystemen wird durch eine übergeordnete Softwareschicht realisiert. Die Interaktionsfolgen werden dazu formal in Form von Prozessmodellen beschrieben und hinterlegt. Diese Beschreibungen können in vielen Fällen flexibel geändert, hierarchisch angeordnet und verschachtelt werden.
Repository ²⁶⁹	In den funktionsbezogenen Bereichen einer Integrationsarchitektur werden vielfältige Metadaten benötigt.	Die Metadaten zur Anwendungsintegration werden in einer zentralen Softwarekomponente verwaltet und anderen Softwarekomponenten der Integrationsarchitektur bereitgestellt.
Database Replication ²⁷⁰	Im Zuge einer informationsorientierten Anwendungsintegration sollen zwischen den Datenbanksystemen der zu integrierenden Anwendungssysteme Daten ausgetauscht werden.	<p>Mittels einer zwischengeschalteten Softwareschicht werden Daten aus dem Datenbanksystem eines Anwendungssystems unmittelbar in das Datenbanksystem eines anderen Anwendungssystems transferiert.</p> <p>Es können verschiedene Varianten dieses Musters unterschieden werden, die sich beispielsweise darin unterscheiden, inwiefern die zwischengeschaltete Softwareschicht auch eine Transformation oder Anreicherung der Daten vornimmt (sog. „extract-transform-load-(ETL)-tool“).</p>

²⁶⁷ Vgl. Linthicum /Application Integration/ 191-215; Lutz /Patterns/ 71f.

²⁶⁸ Vgl. Lutz /Patterns/ 72f.; Linthicum /Application Integration/ 420-422.

²⁶⁹ Vgl. Oey u. a. /SOA/ 157.

²⁷⁰ Vgl. Linthicum /Application Integration/ 7; Kaib /EAI/ 148f.

Muster	Gestaltungsproblem	Lösungsschema
Database Federation ²⁷¹	Im Zuge einer informationsorientierten Anwendungsintegration soll auf die Datenbestände der Datenbanksysteme verschiedener zu integrierender Anwendungssysteme zugegriffen werden können.	Mittels einer zwischengeschalteten Softwareschicht werden die verschiedenen Datenbanksysteme der Anwendungssysteme virtuell zu einer umfassenden Datenbank zusammengefasst. Das heißt, dass eine gemeinsame logische Datenorganisation und zentrale Möglichkeiten zum Lesen und Speichern der Daten in den verschiedenen Datenbanksystemen geschaffen wird.

Tabelle 2-4: Ausgewählte Gestaltungsmuster für Integrationsarchitekturen

Den Gestaltungsmustern für Integrationsarchitekturen liegen insbesondere zwei gemeinsame Anliegen zugrunde. Zum einen sollen die Interaktionen zwischen den Anwendungssystemen möglichst ohne verändernde oder erweiternde Eingriffe in die Anwendungssysteme realisiert werden. Zum anderen sollen die Anwendungssysteme möglichst nicht paarweise durch sog. Punkt-zu-Punkt-Verbindungen verbunden werden.²⁷² Bei n Anwendungssystemen und dem extremen Fall, dass jedes System mit allen anderen Systemen interagieren soll, würde dieses Vorgehen $(n \cdot (n-1))/2$ individuell herzustellende und zu pflegende Verbindungen erfordern.²⁷³ Demgegenüber soll im Zuge der Anwendungsintegration eine gemeinsame Integrationsinfrastruktur geschaffen werden, an welche die Anwendungssysteme flexibel angebunden werden können. Im Idealfall sind für die Integration von n Anwendungssystemen somit nur n Anbindungen vorzunehmen und zu pflegen.²⁷⁴

Daraus folgend stehen die Gestaltungsmuster dafür, jeweils bestimmte funktions- oder qualitätsbezogene Gestaltungsmerkmale der Integrationslösung in zwischengeschalteten und gemeinsam nutzbaren Softwarekomponenten umzusetzen. Diese Maßnahme beinhaltet insbesondere den Einsatz sog. „Middleware“.

Linthicum definiert Middleware umfassend als „jegliche Art von Software, welche die Kommunikation zwischen zwei oder mehr Softwaresystemen unterstützt.“²⁷⁵ Ausge-

²⁷¹ Vgl. Linthicum /Application Integration/ 7-9; Andersson, Johnson /Integration Styles/ 231.

²⁷² Vgl. Kaib /EAI/ 68f.

²⁷³ Vgl. Andersson, Johnson /Integration Styles/ 230. In der Praxis ist dieser extreme Fall jedoch unüblich. In der Regel ist nur ein Teil aller möglichen Punkt-zu-Punkt-Verbindungen sinnvoll (vgl. Boles, Friebe, Luhmann /Integrationsszenarien/ 62).

²⁷⁴ Vgl. Linthicum /EAI/ 3; Kaib /EAI/ 2f.

²⁷⁵ „Middleware is any type of software that facilitates communication between two or more software systems.“ (Linthicum /Application Integration/ 116).

hend von diesem Begriffsverständnis werden mehrere Kategorien von Middleware unterschieden, die spezifische Aspekte der erläuterten Interaktionsformen, der funktions- und qualitätsbezogenen Gestaltungsmerkmale abdecken und bestimmte Gestaltungsmuster repräsentieren. Einige Kategorien von Middleware sind beispielhaft in Tabelle 2-5 erläutert.²⁷⁶

Kategorie von Middleware	Erläuterung
Remote Procedure Call (RPC)	Dieser älteste Typ von Middleware basiert auf dem Interaktionsmechanismus des Prozeduraufrufs. Er ermöglicht einem Programm den Aufruf einer Funktion, die in einem anderen Programm auf einem anderen Rechner ausgeführt wird. Dabei basiert er auf dem synchronen Kommunikationsmodell. ²⁷⁷
Message-Oriented Middleware (MOM)	MOM unterstützt die Kommunikation zwischen zwei oder mehr Softwaresystemen mittels Nachrichten auf der Grundlage des asynchronen Kommunikationsmodells. ²⁷⁸
Distributed Components/Objects	Dieser Typ von Middleware unterstützt die Kommunikation zwischen verteilten Komponenten bzw. den Aufruf verteilter Objekte. ²⁷⁹
Database-Oriented Middleware	Dieser Typ bezeichnet jede Art von Middleware, die die Kommunikation mit Datenbanken unterstützt, sei es aus einem Anwendungssystem heraus oder zwischen mehreren Datenbanken. ²⁸⁰
Transaction-Oriented Middleware	Dieser Typ von Middleware unterstützt und koordiniert die Interaktion zwischen mehreren Softwaresystemen unter Wahrung des Transaktionsprinzips. ²⁸¹

Tabelle 2-5: Ausgewählte Kategorien von Middleware

²⁷⁶ Einen Überblick über die verschiedenen Middleware-Technologien bietet ebenfalls Britton /IT-Architecture/ 19-90.

²⁷⁷ Vgl. Linthicum /Application Integration/ 125f.

²⁷⁸ Vgl. Linthicum /Application Integration/ 126-128.

²⁷⁹ Vgl. Linthicum /Application Integration/ 128.

²⁸⁰ Vgl. Linthicum /Application Integration/ 129f.

²⁸¹ Vgl. Linthicum /Application Integration/ 130-133.

2.2.4.2 Serviceorientierung

Neben den erläuterten Gestaltungsmustern gewinnt ein Entwicklungsparadigma seit einiger Zeit kontinuierlich an Bedeutung: das der serviceorientierten Architekturen (SOA).²⁸² Die Umsetzung des serviceorientierten Entwicklungsparadigmas im Zuge der Anwendungsintegration ist mit einer charakteristischen Gestaltung der Integrationsarchitektur verbunden.

Serviceorientierte Architekturen zeichnen sich dadurch aus, dass Funktionen in Form von wiederverwendbaren, in sich geschlossenen und lose gekoppelten Diensten (engl. services) implementiert werden.²⁸³ Diese Softwarekomponenten können unabhängig von den zugrunde liegenden Implementierungen über wohldefinierte und veröffentlichte Schnittstellen aufgerufen werden.²⁸⁴

Der Aufruf eines Dienstes erfolgt dynamisch, d. h. seine Implementierung wird erst zur Laufzeit bestimmt und dem Aufruf zugeordnet.²⁸⁵ Dazu werden Informationen zu den Diensten in einem Verzeichnis hinterlegt, dem sog. Service-Repository. Das Auffinden eines konkreten Dienstes wird darauf aufbauend durch einen Verzeichnisdienst, die sog. Service-Registry, unterstützt. Eine Softwarekomponente (dies kann ebenfalls ein Dienst sein) sendet eine Anfrage zu einem gewünschten Dienst an den Verzeichnisdienst, der daraufhin dessen Lokalisierung und die Eigenschaften seiner Schnittstelle im Service-Repository feststellt und zurück liefert. Anhand dieser Informationen erfolgt daraufhin der Aufruf des Dienstes. In serviceorientierten Architekturen wird häufig ein sog. Service-Bus eingesetzt. Dieser bildet eine gemeinsame Infrastruktur für die Kommunikation zwischen Diensten und kann beispielsweise Aufgaben der Verschlüsselung, Sicherheitsprüfung, Authentifizierung oder Lastverteilung übernehmen.²⁸⁶

²⁸² Vgl. Linthicum /Application Integration/ 77f. Beispielsweise geht die Meta Group in einer Studie davon aus, dass serviceorientierte Architekturen eine bedeutende Rolle für die zukünftige Entwicklung der Anwendungsintegration spielen werden (vgl. N. a. /META Trends 2004-05/ 1f.).

²⁸³ Als in sich geschlossenen Dienst (self-contained service) bezeichnet man einen Dienst, der für sich allein funktionsfähig ist und einen eigenen Zustand verwaltet (vgl. Hansen, Neumann /Wirtschaftsinformatik/ 783).

²⁸⁴ Vgl. Oey u. a. /SOA/ 149-158.

²⁸⁵ Vgl. Hansen, Neumann /Wirtschaftsinformatik/ 783.

²⁸⁶ Vgl. Oey u. a. /SOA/ 149-158.

Die Entscheidung für eine „serviceorientierte Anwendungsintegration“²⁸⁷ bedeutet, dass die Datenbestände oder Funktionen der zu integrierenden Anwendungssysteme auf geeignete Weise in Form von Diensten erschlossen werden müssen. Linthicum weist darauf hin, dass dies i. d. R. ein stärker invasives Vorgehen erfordert.²⁸⁸ Es ist jedoch nicht immer ein Eingriff in die Anwendungssysteme notwendig. Alternativ kann ein Dienst auf einer bestehenden externen Schnittstelle eines Anwendungssystems aufsetzen und diese erschließen.

Darauf aufbauend unterstützt das serviceorientierte Entwicklungsparadigma die Umsetzung einer geschäftsprozessorientierten Anwendungsintegration. Einzelne Dienste können flexibel im Sinne von Prozessbausteinen verknüpft werden (sog. Orchestrierung) und somit automatisierte Interaktionsfolgen realisiert werden (sog. Choreographien). Die Definition dieser Interaktionsfolgen erfolgt anhand maschinell interpretierbarer Prozessausführungssprachen. Ihre Abwicklung wird daraufhin durch eine zentrale Komponente der Integrationsarchitektur gesteuert.²⁸⁹

Insbesondere für die zwischenbetriebliche Anwendungsintegration wird eine bestimmte Variante des serviceorientierten Entwicklungsparadigmas angewandt, die der Web-Services. Unter Web-Services versteht man lose gekoppelte, verteilte Dienste, die auf der Grundlage von Internettechnologien in einer serviceorientierten Architektur veröffentlicht, lokalisiert und dynamisch aufgerufen werden können.²⁹⁰

Eine zentrale Rolle nimmt dabei die Extensible Markup Language (XML) ein. Diese standardisierte Auszeichnungssprache ermöglicht die Beschreibung der Struktur und Semantik textbasierter Nachrichten, die sowohl maschinell interpretiert als auch durch Menschen gelesen werden können. Eine weitere Stärke von XML resultiert aus ihrem Wesen als Metasprache. Sie ermöglicht es damit, eine Vielfalt von Auszeichnungssprachen für spezielle Zwecke zu definieren. XML bzw. unterschiedliche standardisierte XML-Derivate werden als Nachrichtenprotokolle für die Kommunikation mit Web-Services und Verzeichnisdiensten (Simple Object Access Protocol, SOAP), als Beschreibungssprache für die Schnittstellen von Web-Services (Web-Service Description

²⁸⁷ Linthicum /Application Integration/ 77.

²⁸⁸ Vgl. Linthicum /Application Integration/ 77.

²⁸⁹ Vgl. Ließmann, Wetzke /BPEL und BPMN/ 208-226.

²⁹⁰ Vgl. Hansen, Neumann /Wirtschaftsinformatik/ 803.

Language, WSDL) und als Prozessausführungssprachen für die Choreographie von Web-Services eingesetzt (z. B. Business Process Execution Language, BPEL).²⁹¹

2.2.4.3 Orientierung an Standards

Die Entwicklung von Integrationslösungen ist durch die Orientierung an Standards geprägt. Diese definieren Regelungen zur Gestaltung konzeptioneller und technischer Aspekte der Integrationsarchitekturen. Werden diese Regelungen durchgängig eingehalten, so bedeutet dies das Vermeiden potenzieller Heterogenität. Dadurch wird beispielsweise die Interoperabilität der verschiedenen Softwarekomponenten einer Integrationsarchitektur ebenso wie der Einsatz von Entwicklungswerkzeugen und die Wiederverwendung von Programmcode unterstützt.²⁹² Auch das Qualitätsmerkmal der Flexibilität wird durch das Einhalten von Standards im Allgemeinen positiv beeinflusst.²⁹³

Der Begriff des Standards wird im Kontext der Softwareentwicklung für verschiedene Arten von Gestaltungsregelungen verwendet, die sich im Hinblick auf ihren Ursprung unterscheiden. Einem engen Begriffsverständnis zufolge werden solche Gestaltungsregelungen als Standards bezeichnet, die durch Institutionen, wie z. B. Gremien, Konsortien oder Verbände, im Zuge eines Verhandlungsprozesses zwischen unterschiedlichen Interessensgruppen erarbeitet werden.²⁹⁴ Finden Gestaltungsansätze einzelner Unternehmen, beispielsweise eines Softwareherstellers, infolge der Ausübung von Marktmacht oder der Wirkung von Netzwerkeffekten weite Verbreitung, so spricht man von De-facto-Standards.²⁹⁵ In manchen Fällen werden De-facto-Standards von einer standardisierenden Institution aufgegriffen und als Standard im engen Sinne weiterentwickelt. Ein De-jure-Standard schließlich betrifft Gestaltungsregelungen, die durch eine übergeordnete staatliche Instanz definiert und verbindlich vorgeschrieben werden.²⁹⁶

Es existiert eine Vielzahl von Standards, welche die Gestaltung konzeptioneller und technischer Aspekte von Integrationsarchitekturen regeln. Einen systematischen Über-

²⁹¹ Vgl. Linthicum /Application Integration/ 235-291.

²⁹² Vgl. Buxmann /Standards/ 434.

²⁹³ Vgl. Chari, Seshadri /Integration/ 59.

²⁹⁴ Vgl. Buxmann /Standards/ 434. Eine beispielhafte Aufführung solcher Institutionen findet sich in Ließmann /Schnittstellenorientierung/ 34 und der dort angeführten Quelle.

²⁹⁵ Vgl. Buxmann /Standards/ 434.

²⁹⁶ Vgl. Buxmann /Standards/ 434f.

blick über verschiedene Standards im Kontext der Anwendungsintegration liefern beispielsweise Chari und Seshadri, jedoch ohne sie im Einzelnen zu erläutern. Dabei ordnen sie die Standards nach den Charakteristika der Gestaltungsaspekte, die sie regeln: zum einen danach, ob sie branchenspezifisch oder branchenneutrale Regelungen definieren, zum anderen danach, welche der drei bereits erläuterten Interaktionsschichten der Anwendungssysteme sie betreffen (Benutzerschnittstelle, Anwendungsfunktionalität, Datenverwaltung) und schließlich danach, welche der funktionsbezogenen Gestaltungsmerkmale einer Integrationslösung sie betreffen, wobei sich die Darstellung auf die Übermittlung von Informationen, die Transformation und Anreicherung von Informationen und die Interaktionsauslösung und -steuerung beschränkt.²⁹⁷

Diese Vielzahl von Standards bedeutet jedoch nicht, dass die Entwicklung einer Integrationsarchitektur vollständig und einheitlich geregelt und das Treffen von Gestaltungsentscheidungen kaum mehr erforderlich wäre. Auf der einen Seite definieren mehrere Standards unterschiedliche Regelungen für denselben Gestaltungsaspekt, auf der anderen Seite werden bestimmte Gestaltungsaspekte durch keine existierenden Standards geregelt.²⁹⁸

Neben ihrer Zuordnung zu den Gestaltungsaspekten, die sie regeln, können die Standards anhand weiterer Merkmale charakterisiert werden, die ihren Einsatz und die damit verbundenen Auswirkungen im Rahmen der Entwicklung einer Integrationsarchitektur beeinflussen. Einige dieser Merkmale werden in Tabelle 2-6 erläutert.

²⁹⁷ Vgl. Chari, Seshadri /Integration/ 61-63. Eine systematische Darstellung von Standards zur Anwendungsintegration bieten auch Voigtmann, Zeller /Integrationssysteme/ 2-6 oder Ließmann /Schnittstellenorientierung/ 33-50.

²⁹⁸ Vgl. Chari, Seshadri /Integration/ 59; Buxmann /Standards/ 434.

Merkmal	Erläuterung
Detaillierungsgrad	Standards unterscheiden sich hinsichtlich des Detaillierungsgrades ihrer Spezifikation bzw. der Freiheitsgrade welche sie bei ihrer Umsetzung einräumen. Während eine geringe Detaillierung der Regelungen aus Gründen der Einfachheit ihrer Beschreibung in manchen Fällen als angemessen erscheinen kann, kann sie in anderen Fällen dazu führen, dass die genannten Ziele der Standardisierung nicht erreicht werden. ²⁹⁹
Technologieneutralität	Oftmals wird bei der Definition von Standards angestrebt, diese unabhängig von konkreten technischen Implementierungen festzulegen und somit alternative technische Realisierungsmöglichkeiten zu ermöglichen. ³⁰⁰ Dies trifft jedoch nicht auf alle Standards zu, insbesondere dann nicht, wenn auch De-facto-Standards in die Betrachtung einbezogen werden, die meist auf konkreten Technologien bestimmter Hersteller basieren. Aber auch andere Standards können die Auswahl der Technologie einschränken, beispielsweise unmittelbar, indem sie den Einsatz bestimmter Technologien regeln, oder mittelbar, indem sie auf bestimmten Paradigmen basieren, die nur von bestimmten Technologien unterstützt werden.
Stabilität	Standards und ihre Spezifikationen werden im Laufe der Zeit meist weiterentwickelt. Werden dadurch Anpassungen an den Implementierungen früherer Entwicklungsstände erforderlich, um die Kompatibilität zu gewährleisten, so bedeutet dies eine Einschränkung der Vorteile der Verwendung der Standards im Hinblick auf die Flexibilität und Änderbarkeit sowie die Wiederverwendbarkeit einer Integrationsarchitektur. ³⁰¹ Die Stabilität eines Standards im Hinblick auf die Änderungsdynamik und der Kompatibilität zu früheren Entwicklungsständen ist damit ein weiteres wichtiges Merkmal der Standards.
Verbreitung	Die Orientierung an Standards ist mit Netzeffekten verbunden. D. h. der Nutzen eines Standards zur Vermeidung von Homogenität ebenso wie die Verfügbarkeit von Entwicklungswerkzeugen und wieder verwendbarem Programmcode steigt im Allgemeinen mit seiner Verbreitung an. ³⁰² Mit zunehmender Verbreitung eines Standards treten damit positive Rückkopplungseffekte auf, die ihm weiteres Gewicht verleihen.
Kompatibilität	Einzelne Standards regeln jeweils nur einen Ausschnitt der Gestaltungsaspekte einer Integrationsarchitektur. I. d. R. kommen daher mehrerer Standards im Rahmen einer Integrationsarchitektur zum Einsatz. Dabei sind die verschiedenen Standards in unterschiedlichem Maße kompatibel zu einander, wodurch ihre prinzipielle Kombinierbarkeit bzw. das Ausmaß erforderlicher Anpassungen oder Maßnahmen zur Auflösung von Konflikten determiniert werden. ³⁰³

Tabelle 2-6: Merkmale von Standards für die Gestaltung von Integrationsarchitekturen

²⁹⁹ Vgl. Buxmann /Standards/ 435; Chari, Seshadri /Integration/ 59.

³⁰⁰ Vgl. Gierharke /Geschäftsprozessmanagement/ 83f.

³⁰¹ Vgl. Chari, Seshadri /Integration/ 59f.

³⁰² Vgl. Buxmann /Standards/ 434.

³⁰³ Vgl. Chari, Seshadri /Integration/ 59.

2.2.4.4 Einsatz von Integrationsprodukten

Wie erläutert, umfasst eine Integrationslösung i. d. R. verschiedene Softwarekomponenten, welche die Interaktionen der Anwendungssysteme unterstützen. Diese Bestandteile einer Integrationsarchitektur werden im zunehmenden Maße nicht mehr individuell entwickelt, sondern als vorgefertigte Komponenten erworben. In Folge dieses Trends hat sich ein dynamischer Markt herausgebildet, auf dem Komponenten für die Realisierung von Integrationsarchitekturen und mit ihnen zusammenhängende Dienstleistungen bezogen werden können. Diese Standardsoftwaresysteme werden im Weiteren als Integrationsprodukte bezeichnet.³⁰⁴

Mit dem Einsatz von Integrationsprodukten werden, wie generell beim Einsatz von Standardsoftwarekomponenten, im Allgemeinen ein geringerer Entwicklungsaufwand und eine kürzere Entwicklungsdauer sowie eine höhere Qualität der Integrationslösung verbunden. Die gleiche Motivation liegt auch der Wiederverwendung von Software zugrunde. Der Einsatz von Integrationsprodukten unterscheidet sich jedoch üblicherweise in dreierlei Hinsicht von der reinen Wiederverwendung von Software:³⁰⁵

1. Es kann angenommen werden, dass Integrationsprodukte selbständig lauffähig sind, während dies bei wieder verwendeter Software nicht der Fall sein muss.
2. Wieder verwendete Software stammt im Allgemeinen von der Organisation, die das Entwicklungsvorhaben durchführt. Integrationsprodukte hingegen stammen von unabhängigen Herstellern.
3. Die Wiederverwendung von Software erfordert im Allgemeinen den Zugriff auf deren Quellcode, während der Quellcode der lizenzierten Integrationsprodukte oftmals nicht zugänglich ist.

Die Verwendung von Integrationsprodukten bringt damit weitere Umstände mit sich: Man hat keine Kontrolle über die funktions- und qualitätsbezogenen Merkmale der Integrationsprodukte. Man hat keine Kontrolle über ihre Evolution. Zudem kann das Verhalten der Hersteller, beispielsweise im Hinblick auf den von ihnen angebotenen Support, sehr unterschiedlich sein.³⁰⁶

³⁰⁴ Vgl. Kaib /EAI/ 134.

³⁰⁵ Vgl. Abts, Boehm /COCOTS/ 13.

³⁰⁶ Vgl. Abts, Boehm /COCOTS/ 69.

Integrationsprodukte realisieren bestimmte Gestaltungsmuster für Integrationsarchitekturen und repräsentieren oder basieren im Allgemeinen auf bestimmten Kategorien von Middleware. Zur vollständigen Lösung eines konkreten Integrationsproblems ist häufig der kombinierte Einsatz verschiedener Integrationsprodukte erforderlich, oftmals ergänzt um individuell entwickelte Softwarekomponenten.

Viele Hersteller haben in den letzten Jahren den funktionalen Umfang der Integrationsprodukte ausgeweitet. Diese unterstützen somit zunehmend mehrere funktionsbezogene Gestaltungsbereiche einer Integrationsarchitektur. Zum anderen haben viele Hersteller ihr Sortiment an Integrationsprodukten erweitert, so dass sie umfangreiche Integrationslösungen unterstützen können. Es zeichnet sich ab, dass auch in den kommenden Jahren die Produkte mehr und mehr zusammenwachsen und sich die Anzahl bedeutsamer Hersteller konsolidieren wird.³⁰⁷

Verschiedene Studien stellen als Momentaufnahme verschiedene Ausschnitte des Marktes für Integrationsprodukte dar.³⁰⁸ Als Beispiel kann diesbezüglich der „Magic Quadrant for Application Integration Suites“ von Gartner Inc. genannt werden.³⁰⁹ In dieser Studie werden Hersteller von Integrationsprodukten im Hinblick auf die Vollständigkeit ihrer Vision, im Sinne der Unterstützung aller funktionsbezogener Merkmale einer Integrationsarchitektur, sowie im Hinblick auf ihre Fähigkeit zur Umsetzung dieser Vision, im Sinne ihrer wettbewerbsstrategischen Position, bewertet. Anhand dieser Bewertung werden die Hersteller in vier Quadranten unterteilt, die ihre Position im Markt für Integrationsprodukte charakterisieren sollen.

³⁰⁷ Vgl. Boston Corporate Finance/ Enterprise Integration/ 15; Kaib /EAI/ 154f.

³⁰⁸ Vgl. beispielsweise Buhl, Christ, Pape /Marktstudie/ 43-93; Boston Corporate Finance/ Enterprise Integration/ 14-23.

³⁰⁹ Vgl. Thompson u. a. /Application Integration Suites 2Q04/ 3f.

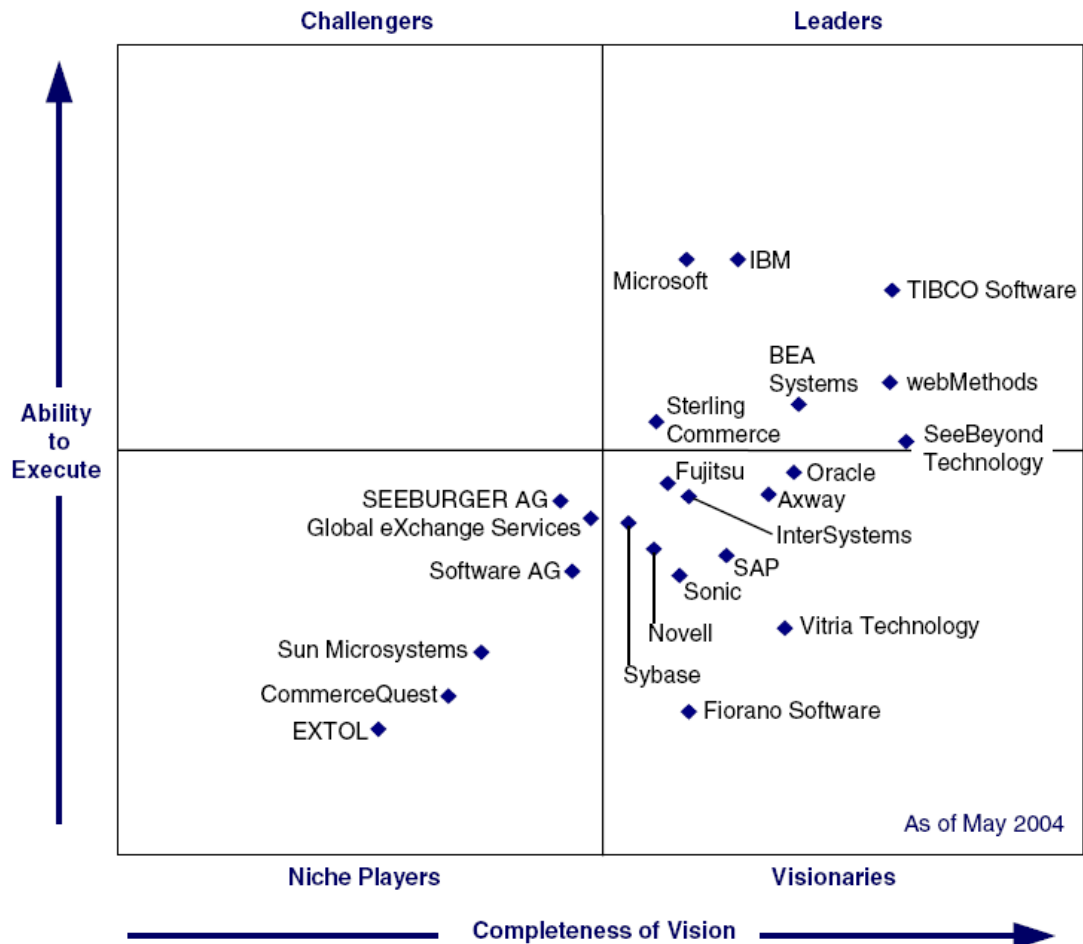


Abbildung 2-7: Magic Quadrant for Application Integration Suites, 2. Quartal 2004³¹⁰

Die verschiedenen Integrationsprodukte zeichnen sich durch spezifische Charakteristika aus, die maßgeblich ihre Eignung für die Lösung eines konkreten Integrationsproblems beeinflussen. Zur Unterstützung einer systematischen Evaluierung und Auswahl von Integrationsprodukten im Rahmen von Integrationsvorhaben werden verschiedene Modelle vorgeschlagen. Diese grenzen in unterschiedlicher Weise Kriterien ab, anhand derer die Integrationsprodukte charakterisiert und den aus einem konkreten Integrationsproblem resultierenden Anforderungen gegenübergestellt werden können. Neben den Integrationsprodukten sind es die jeweiligen Hersteller selbst, die unter strategischen und taktischen Gesichtspunkten im Rahmen der Modelle zu beurteilen sind. Als Beispiele können das EAI-Produktmodell des Marktforschungsunternehmens Ovum Ltd. oder die „Giga-Scorecard“ zur Beurteilung von Integrationsprodukten der Giga Information Group genannt werden.³¹¹ Um eine fundierte Auswahlentscheidung treffen

³¹⁰ Thompson u. a. /Application Integration Suites 2Q04/ 2.

³¹¹ Vgl. Gilpin /EAI Solution/ 2-4; Buhl, Christ, Pape /Marktstudie/ 37-41; Kaib /EAI/ 161-166.

zu können, ist es im Allgemeinen erforderlich, die genannten Modelle situationsspezifisch anzupassen und zu erweitern.³¹²

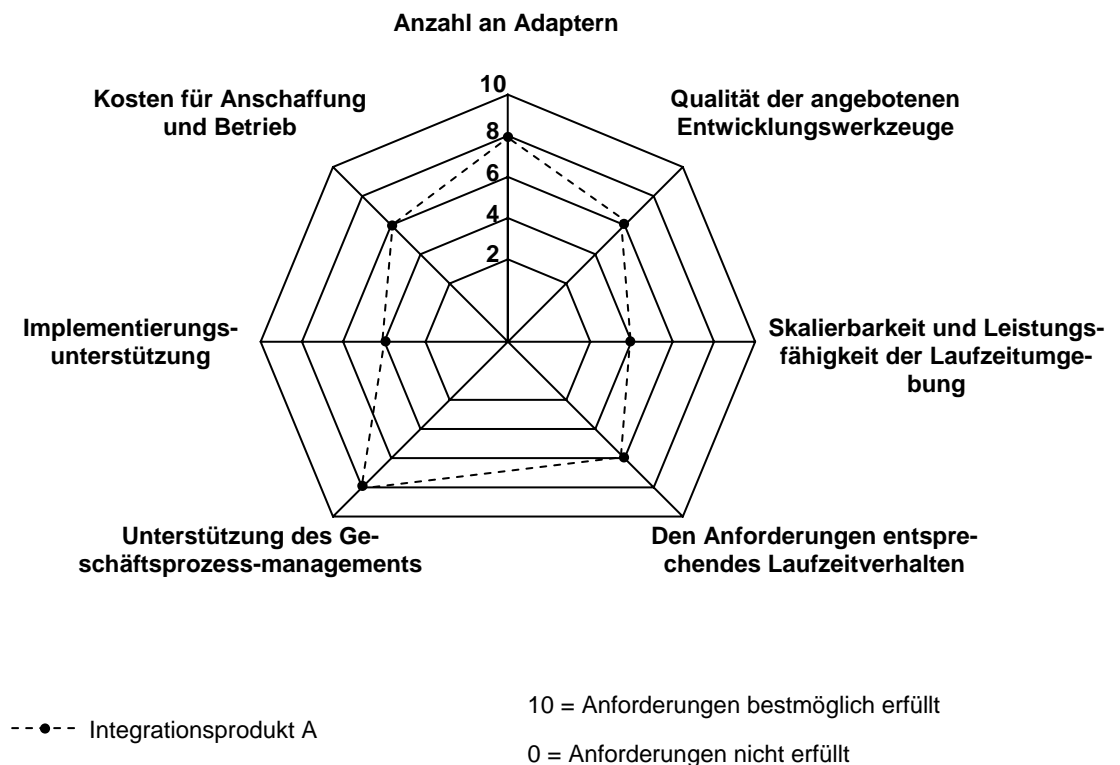


Abbildung 2-8: „Giga-Scorecard“ zur Beurteilung von Integrationsprodukten³¹³

Wenn ein Integrationsprodukt zur Lösung eines konkreten Integrationsproblems eingesetzt werden soll, so kann in unterschiedlicher Form der Bedarf oder die Möglichkeit einer Anpassung an problemspezifische Gegebenheiten bestehen. Zum einen können alternative Konfigurationen des Integrationsprodukts ermöglicht werden. Zum anderen ist es häufig erforderlich, spezifische Informationen zur Unterstützung der jeweiligen funktionsbezogenen Bereiche der Integrationsarchitektur in Form von Daten, Algorithmen oder Modellen zu hinterlegen.

Die natürlichen Grenzen der Einsatzmöglichkeiten von Integrationsprodukten bestehen dort, wo die Vielfalt der zu erfüllenden Anforderungen die Entwicklung vorgefertigter Komponenten oder deren wirtschaftlichen Vertrieb verhindert. Aus diesem Grund ist der Einsatz von Integrationsprodukten auch längerfristig oftmals mit einem Restbedarf an individueller Entwicklung von Softwarekomponenten verbunden, um die Erfüllung

³¹² Vgl. Gilpin /EAI Solution/ 6.

³¹³ In Anlehnung an Kaib /EAI/ 163.

der spezifischen funktionsbezogenen und qualitätsbezogenen Anforderungen zu erreichen. Die Notwendigkeit der individuellen Entwicklung tritt insbesondere an den Schnittstellen der einzelnen Softwarekomponenten einer Integrationslösung auf. Es ist dort vielfach erforderlich, Programmcode einzufügen, der die Interaktion der Komponenten ermöglicht und der oftmals als „glue-ware“ oder „glue-code“ bezeichnet wird. Angesichts dieser Umstände besitzen viele Integrationsprodukte integrierte Entwicklungswerkzeuge, die eine Ergänzung um individuell entwickelte Softwarekomponenten und die Verknüpfung der Softwarekomponenten unterstützen.³¹⁴

2.3 Integrationsvorhaben

Aufbauend auf diesem Verständnis der Integrationsprobleme und der Integrationslösungen werden im Weiteren die Charakteristika der Integrationsvorhaben betrachtet. Insbesondere erscheint relevant, welche Aufgaben auf dem Weg vom Integrationsproblem zur Integrationslösung erfüllt werden müssen, denn sie erfordern personelle Arbeitsleistung, d. h. sie sind mit Aufwand verbunden.

Zur Beschreibung von Integrationsvorhaben existieren verschiedene Ansätze, beispielsweise von Edwards und Newing, von Lam und Shankararaman, von Linthicum, von Roch, von Ruh, Maginnis und Brown, von Schmietendorf u. a. sowie von Yee und Apte.³¹⁵ Diese können jedoch lediglich als Ausgangspunkte für das Verständnis der Besonderheiten von Integrationsvorhaben angesehen werden. Von Vorgehensmodellen kann bei den angeführten Beiträgen kaum gesprochen werden.³¹⁶ Die einzelnen Teilaufgaben und organisatorischen Gestaltungsmaßnahmen der Integrationsvorhaben werden (auch innerhalb der einzelnen Beiträge) unterschiedlich detailliert erläutert, auf einige Aspekte wird gar nicht eingegangen, an anderer Stelle wiederum wird explizit auf Konzepte der klassischen Softwareentwicklung verwiesen, die gemäß den Autoren auch für die Anwendungsintegration gültig seien. Darüber hinaus bleibt der Ursprung der von

³¹⁴ Vgl. Lee, Siau, Hong /Enterprise Integration/ 57f.; Kurbel /Software-Engineering/ 495f.; Kaib/EAI/ 164; Abts, Boehm /COCOTS/ 67.

³¹⁵ Vgl. Edwards, Newing /Application Integration/ 197-203; Lam, Shankararaman /Enterprise Integration/ 41-48; Linthicum /Application Integration/ 370-392; Roch /Methodology for EAI – Part I/ und Roch /Methodology for EAI – Part II/; Ruh, Maginnis, Brown /EAI/ 153-179; Schmietendorf u. a. /EAI/; Yee, Apte /e-Business Enterprise/ 205-221.

³¹⁶ Es verbleiben vielfach Unklarheiten, beispielsweise bezüglich des jeweiligen Gegenstands der Teilaufgaben, der jeweils erforderlichen Vor- und Nachbedingungen, der zwischen den Teilaufgaben bestehenden Beziehungen oder die Zuordnung der Teilaufgaben zu Aufgabenträgern im Sinne eines Rollenmodells. Zum Begriff des Vorgehensmodells siehe Seibt /Vorgehensmodell/.

den Autoren präsentierten Erkenntnisse unklar. Da sie nur unzureichend auf zugrunde gelegte Quellen verweisen, scheint es sich vorwiegend um ihre persönlichen Erfahrungen und Einschätzungen zu handeln.³¹⁷ Letztlich spiegelt sich darin die der Arbeit zugrunde liegende Problemstellung wider: Es liegen noch zu wenige Erkenntnisse zu den Besonderheiten der Anwendungsintegration vor.

Um zu einem besseren Verständnis der Integrationsvorhaben zu gelangen, erscheint es daher angemessen, etablierte Konzepte der Softwareentwicklung im Sinne eines generischen Modells aufzugreifen, um darauf aufbauend die in den o. g. Beiträgen beschriebenen Besonderheiten von Integrationsvorhaben zu erläutern.

2.3.1 Generisches Modell von Entwicklungsvorhaben

Im Allgemeinen kann Softwareentwicklung als Aufgabe verstanden werden, eine zur Lösung eines Problems geeignete Software zu entwickeln und bereitzustellen. Ein Entwicklungsvorhaben besitzt damit das Wesen der Lösung eines Entscheidungsproblems. Das zu lösende Problem muss identifiziert und verstanden werden. Wählbare Lösungsalternativen müssen bestimmt und bewertet werden und die Auswahl einer Alternative getroffen werden. Dieser Entschluss muss anschließend realisiert werden.³¹⁸ In diesem Sinne kann ein Entwicklungsvorhaben verrichtungsorientiert in verschiedene Teilaufgaben untergliedert werden, die jeweils einzelne der genannten Schritte zur Bewältigung der Gesamtaufgabe betreffen und als Entwicklungsaufgaben bezeichnet werden.³¹⁹

Die Teilaufgabe Analyse betrifft die Problembestimmung und darauf aufbauend die Definition der externen, d. h. der durch den Benutzer wahrnehmbaren Merkmale der Software.³²⁰ Die Teilaufgabe Entwurf betrifft die Konzeption der internen Merkmale der Software.³²¹ Die Teilaufgabe Implementierung betrifft die Realisierung der entwor-

³¹⁷ Lediglich Linthicum sowie Schmietendorf u. a. führen abschließend ein Verzeichnis referenzierter Literatur an. Da sie jedoch im Rahmen ihrer Ausführungen nicht explizit auf zugrunde liegende Quellen verweisen, sind potenzielle Bezugnahmen nicht ohne weiteres nachvollziehbar.

³¹⁸ Vgl. Witte /Entscheidungsprozesse/ 552-554; Putz-Osterloh /Entscheidungsverhalten/ 585f.

³¹⁹ Vgl. Mellis /Projektmanagement/ 63; Stelzer /Software-Qualitätsmanagement/ 98f.

³²⁰ Vgl. Mellis /Projektmanagement/ 65f.

³²¹ Die verbreitete Untergliederung in Grobentwurf und Feinentwurf bezeichnet den zunehmend hohen Detaillierungsgrad der Konzeption, von der Festlegung der Architektur der Software hin zur Festlegung einzelner Funktionen, Daten- und Kontrollstrukturen oder Benutzerschnittstellen. Vgl. Mellis /Projektmanagement/ 66f.

fenen Software.³²² Die Teilaufgabe Überführung in die Nutzung schließlich bezeichnet die Bereitstellung der implementierten Software für die Benutzer.³²³

Neben diesen Entwicklungsaufgaben können verschiedene Unterstützungsaufgaben unterschieden werden.³²⁴ Sie sollen die zielführende Erfüllung der Entwicklungsaufgaben unterstützen und die zuverlässige Umsetzung der getroffenen Entwicklungsentscheidungen sicherstellen. Die Unterstützungsaufgaben zeichnen sich durch ihren Querschnittscharakter aus, denn sie sind entwicklungsaufgabenübergreifend zu erfüllen.³²⁵

Die Teilaufgabe Planung umfasst alle in die Zukunft gerichteten Entscheidungen, die einen erfolgreichen Verlauf des Entwicklungsvorhabens vorbereiten und die künftige Erfüllung von Entwicklungsaufgaben im Voraus aufeinander abstimmen sollen. Die Teilaufgabe Kontrolle besteht darin, sicherzustellen, dass der tatsächliche Verlauf des Entwicklungsvorhabens mit dem geplanten Verlauf übereinstimmt. Die Teilaufgabe Qualitätssicherung ist darauf ausgerichtet, die Erfüllung von Anforderungen an die zu entwickelnde Software und den Entwicklungsvorgang zu überprüfen, um nötigenfalls Korrekturmaßnahmen einzuleiten. Die Teilaufgabe Konfigurationsmanagement umfasst die Identifikation und Verwaltung aller Arbeitsergebnisse der Softwareentwicklung sowie ihrer Konfigurationen. Die Teilaufgabe Dokumentierung schließlich betrifft die Erstellung und Verwaltung von Aufzeichnungen über Arbeitsergebnisse und den Entwicklungsvorgang.³²⁶

Die Erfüllung der unterschiedenen Entwicklungs- und Unterstützungsaufgaben übersteigt in aller Regel die Möglichkeiten einer einzelnen Person. Die Teilaufgaben werden vielmehr arbeitsteilig bearbeitet, d. h. sie werden verschiedenen Aufgabenträgern zugeordnet, wobei sie aufgrund ihres Umfangs i. d. R. weiter zerlegt werden müssen.³²⁷ Da-

³²² Vgl. Mellis /Projektmanagement/ 67.

³²³ Vgl. Mellis /Projektmanagement/ 67.

³²⁴ Vgl. Mellis /Projektmanagement/ 63; Stelzer /Software-Qualitätsmanagement/ 98f.

³²⁵ Vgl. Mellis /Projektmanagement/ 63; Stelzer /Software-Qualitätsmanagement/ 102. Zu den angeführten Unterstützungsaufgaben vgl. Mellis /Projektmanagement/ 68f.

³²⁶ Mellis nennt diese Unterstützungsaufgabe „Dokumentation“. Um Missverständnisse mit dem Begriff der Dokumentation im Sinne einer Aufzeichnung über konzeptionelle oder technische Charakteristika von Softwaresystemen zu vermeiden, wird in dieser Arbeit der Begriff „Dokumentierung“ gewählt.

³²⁷ Dabei ist es durchaus möglich, dass ein Aufgabenträger verschiedene Teilaufgaben bearbeitet.

mit ergibt sich die Notwendigkeit, die zielführende Erfüllung der Aufgabenkomplexe durch organisatorische Gestaltungsmaßnahmen sicherzustellen.³²⁸

Aufbauorganisatorische Maßnahmen betreffen die horizontale und vertikale Zerlegung der Aufgaben, d. h. die Abgrenzung von Aufgabeninhalten und Entscheidungs- bzw. Weisungsbefugnissen, und die Zuweisung der abgegrenzten Aufgabenkomplexe auf organisatorische Einheiten.³²⁹ Ablauforganisatorische Maßnahmen betreffen dagegen die „zeitliche und räumliche Gestaltung der Teilaufgaben und ihre Zusammenfassung zu Prozessen.“³³⁰ Abstimmungsmaßnahmen werden ergriffen, um die Erfüllung der interdependenten Teilaufgaben im Hinblick auf ein übergeordnetes Zielsystem zu koordinieren.³³¹ Gestaltungsmaßnahmen zur Kommunikation betreffen den Austausch von Informationen zwischen den an dem Entwicklungsvorhaben beteiligten organisatorischen Einheiten.³³² Motivationsmaßnahmen schließlich bezwecken die Ausrichtung der individuellen Einstellungs- und Verhaltensmerkmale der Mitarbeiter auf ein gemeinsames Zielsystem.³³³

³²⁸ Vgl. Mellis /Projektmanagement/ 69f.; Stelzer /Software-Qualitätsmanagement/ 104-106.

³²⁹ Vgl. Mellis /Projektmanagement/ 72 bzw. die dort zitierte Quelle.

³³⁰ Mellis /Projektmanagement/ 72.

³³¹ Vgl. Mellis /Projektmanagement/ 71; Stelzer /Software-Qualitätsmanagement/ 110-113.

³³² Vgl. Mellis /Projektmanagement/ 73.

³³³ Vgl. Mellis /Projektmanagement/ 72; Stelzer /Software-Qualitätsmanagement/ 113-116.

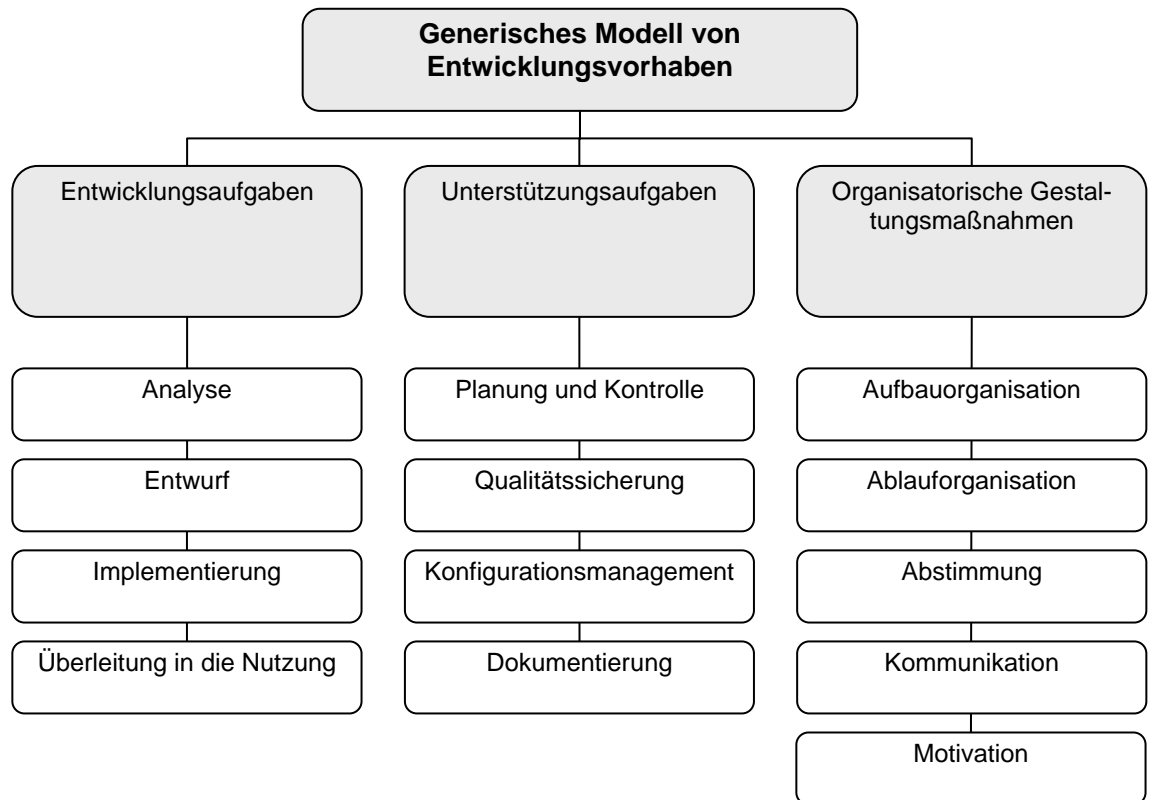


Abbildung 2-9: Generisches Modell von Entwicklungsvorhaben

2.3.2 Besonderheiten der Integrationsvorhaben

Die erläuterten Teilaufgaben sowie die organisatorischen Gestaltungsmaßnahmen können als generische Aspekte von Entwicklungsvorhaben angesehen werden, d. h. sie gelten auch für Integrationsvorhaben. Es wurde bereits erläutert, dass Integrationsvorhaben sowohl charakteristische Problemstellungen als auch charakteristische Lösungen zugrunde liegen. Daraus kann geschlossen werden, dass Besonderheiten im Hinblick auf die Ausprägung der Teilaufgaben und die organisatorischen Gestaltungsmaßnahmen bestehen müssen. Im Weiteren werden die in diesem Zusammenhang vorliegenden Erkenntnisse dargestellt.

▪ Besonderheiten der Analyse

Die Teilaufgabe Analyse betrifft die Problembestimmung, d. h. das Verständnis der an die Software gerichteten Bedürfnisse und Wünsche sowie der bestehenden Restriktionen, und darauf aufbauend die Definition der zu entwickelnden Software, im Sinne der zu erfüllenden Anforderungen. Aus den erläuterten Charakteristika der Integrationsprobleme können Besonderheiten der Analyse im Rahmen von Integrationsvorhaben gefolgert werden.

Ein Integrationsproblem wurde definiert als ein Integrationszustand, der gegenüber einem Integrationsbedarf als strukturell oder qualitativ unzureichend angesehen wird. Dies bedeutet, dass die Problembestimmung zunächst ein Verständnis des Integrationsbedarfs erfordert. Diesbezüglich gilt es, die Strategie und damit die grundlegenden Ziele der Informationsverarbeitung zu bestimmen. Ferner müssen die relevanten Geschäftsprozesse und die in ihnen anfallende Informationsverarbeitung verstanden werden. Schließlich ist es erforderlich, die eingesetzten Anwendungssysteme zu identifizieren und diese den Geschäftsprozessen zuzuordnen, um die Abhängigkeiten zwischen den Anwendungssystemen aufzudecken.³³⁴ Darauf hin muss der bestehende Integrationszustand der Anwendungssysteme erhoben und dem identifizierten Integrationsbedarf gegenüber gestellt werden.³³⁵

Des Weiteren gilt es, die Beschaffenheit der zu integrierenden Anwendungssysteme zu verstehen, d. h. ihre relevanten konzeptionellen und technischen Charakteristika zu bestimmen.³³⁶ Die Dokumentation der Anwendungsintegration stellt in diesem Zusammenhang eine wichtige Informationsquelle dar. Falls keine ausreichende Dokumentation verfügbar ist, so müssen die konzeptionellen und technischen Charakteristika durch die Untersuchung der Softwarekomponenten selbst erarbeitet und auf einer höheren Abstraktionsebene beschrieben werden (Reverse Engineering).³³⁷ Dabei können zwei Ansätze unterschieden werden: Sofern der Quellcode der Anwendungssysteme verfügbar ist, kann dieser untersucht und die benötigten Informationen extrahiert werden („white-box“ Ansatz). Alternativ kann das extern wahrnehmbare Verhalten der Anwendungssysteme, insbesondere die erforderlichen Eingaben und die Ausgaben an den externen Schnittstellen, untersucht werden, um auf die Charakteristika der Anwendungssysteme zu schließen („black-box“ Ansatz).³³⁸

Neben den zu integrierenden Anwendungssystemen müssen womöglich auch frühere Maßnahmen zur Integration der Anwendungssysteme berücksichtigt werden. Dort gilt es ebenfalls, aufgrund der verfügbaren Dokumentation oder der Untersuchung von Softwarekomponenten die relevanten konzeptionellen und technischen Charakteristika zu bestimmen.

³³⁴ Vgl. Edwards, Newing /Application Integration/ 198; Lam, Shankararaman /Enterprise Integration/ 41f.; Yee, Apte /e-Business Enterprise/ 209-211; Linthicum /Application Integration/ 378f., 386.

³³⁵ Vgl. Linthicum /Application Integration/ 370.

³³⁶ Vgl. Linthicum /Application Integration/ 370-378, 384-386.

³³⁷ Vgl. Jung /Integrationsorientiertes Reengineering/ 4.

³³⁸ Vgl. Comella-Dorda /Legacy Systems/ 4f.

Linthicum empfiehlt die in den genannten Tätigkeiten erhobenen Informationen zentral zu dokumentieren und sie laufend auf Vollständigkeit und Korrektheit zu überprüfen. Insbesondere weist er in diesem Zusammenhang auf den Nutzen eines Geschäftsprozesskatalogs, eines umfassenden Datenkatalogs und eines Schnittstellenverzeichnisses hin.³³⁹

Aufbauend auf den erhobenen Informationen hat die Teilaufgabe Analyse die Spezifizierung der herzustellenden Interaktionen der Anwendungssysteme sowie der funktions- und qualitätsbezogenen Anforderungen an die Integrationslösung zum Gegenstand. Bei der Anwendungsintegration steht nicht die Interaktion zwischen einer Software und einem Benutzer, sondern die Interaktion zwischen Anwendungssystemen im Mittelpunkt. Daher sind nicht nur die für den Benutzer wahrnehmbaren Merkmale der Integrationslösung der Teilaufgabe Analyse zuzuordnen, sondern auch die von den Anwendungssystemen erforderten technischen Merkmale.³⁴⁰

▪ **Besonderheiten des Entwurfs**

Die Teilaufgabe Entwurf betrifft die Konzeption der internen Merkmale der zu entwickelnden Software. Im Falle der Anwendungsintegration bezieht sich diese Teilaufgabe auf die zunehmend detaillierte Konzeption der Integrationsarchitektur im Zusammenhang mit der Auswahl der einzusetzenden Technologien und Integrationsprodukte.

Mit der Integrationsarchitektur gilt es die funktions- und qualitätsbezogenen Merkmale der Integrationslösung zu gestalten. Linthicum sowie Ruh, Maginnis und Brown empfehlen angesichts dieser großen Bedeutung der Integrationsarchitektur, auf bewährte Lösungsschemata zurückzugreifen und Gestaltungsmuster zu verwenden.³⁴¹ Die diesbezüglich in der Fachliteratur diskutierten Ansätze wurden in dieser Arbeit bereits vorgestellt.³⁴²

Im Zusammenhang mit der Gestaltung der Integrationsarchitektur steht die Auswahl der einzusetzenden Technologien und Integrationsprodukte. Dabei gilt es diejenigen Tech-

³³⁹ Vgl. Linthicum /Application Integration/ 370-379, 384-386.

³⁴⁰ Vgl. Yee, Apte /e-Business Enterprise/ 211.

³⁴¹ Vgl. Linthicum /Application Integration/ 382-384; Ruh, Maginnis, Brown /EAI/ 172.

³⁴² Vgl. Kapitel 2.2.4.1.

nologien und Integrationsprodukte zu bestimmen, welche eine möglichst hohe Erfüllung der spezifizierten Anforderungen gewährleisten.³⁴³ Neben der Erfüllung der spezifizierten Anforderungen spielen jedoch auch strategische und taktische Aspekte bei der Festlegung auf die Hersteller von Integrationsprodukten eine wichtige Rolle. Die in der Fachliteratur diskutierten Modelle zur systematischen Evaluierung und Auswahl von Integrationsprodukten wurden in dieser Arbeit ebenfalls bereits beispielhaft vorgestellt.³⁴⁴

Es wurde darauf hingewiesen, dass neben dem Einsatz von Integrationsprodukten oftmals ein Restbedarf an individueller Entwicklung von Softwarekomponenten verbleibt.³⁴⁵ Darüber hinaus kann entschieden werden, in die Anwendungssysteme einzugreifen, um bestehende externe Schnittstellen für die beabsichtigte Integration zu modifizieren oder gar neue Schnittstellen zu schaffen.³⁴⁶ Die Konzeption dieser Maßnahmen ist ebenfalls der Teilaufgabe Entwurf zuzuordnen.

Eine Integrationsarchitektur kann aus verschiedenen Perspektiven beschrieben werden (unterschiedliche Sichten). Diesbezüglich empfehlen Yee und Apte eine Integrationsarchitektur aus drei Perspektiven zu definieren: im Hinblick auf das Zusammenwirken der einzelnen Softwarekomponenten der Integrationslösung, die Zuordnung der Komponenten zu verschiedenen Schichten und die physische Verteilung der Integrationslösung.³⁴⁷ Ruh, Maginnes und Brown empfehlen dagegen zehn verschiedene Sichten zur Beschreibung einer Integrationsarchitektur, jedoch ohne diese eingehend zu erläutern oder zu begründen.³⁴⁸ Sie sind in Tabelle 2-7 aufgeführt.

³⁴³ Vgl. Linthicum /Application Integration/ 387.

³⁴⁴ Vgl. Kapitel 2.2.4.4.

³⁴⁵ Vgl. Lee, Siau, Hong /Enterprise Integration/ 57f.; Kurbel /Software-Engineering/ 495f.; Kaib/EAI/ 164; Abts, Boehm /COCOTS/ 67.

³⁴⁶ Vgl. Fincham /Application Integration/ 23.

³⁴⁷ Vgl. Yee, Apte /e-Business Enterprise/ 215-217.

³⁴⁸ Vgl. Ruh, Maginnis, Brown /EAI/ 169.

Perspektive	Erläuterung
Komponenten-Sicht	Diese Sicht beschreibt die Untergliederung der Architektur in Komponenten, die dann fortschreitend in weitere Subkomponenten unterteilt werden.
Entitäten-/Objekt-Sicht	Diese Sicht beschreibt die Geschäftseinheiten, die in Form wiederverwendbarer Komponenten in der Integrationslösung bereitgestellt werden.
Benutzer-Sicht	Diese Sicht beschreibt die Benutzerschnittstelle der Integrationslösung.
Daten-Sicht	Diese Sicht beschreibt die Datenbestände, die im Rahmen der Integrationslösung verwaltet werden.
Altsystem-Sicht	Diese Sicht beschreibt, wie die Altsysteme in die Integrationsarchitektur eingebunden sind.
Sicherheits-Sicht	Diese Sicht beschreibt die unterstützten Sicherheitskonzepte und die Komponenten, welche sie realisieren.
Physische Sicht	Diese Sicht beschreibt, wie Software und Daten der Integrationslösung auf Hardwarekomponenten verteilt sind.
Systemmanagement-Sicht	Diese Sicht beschreibt, wie die einzelnen Bestandteile der Integrationsarchitektur überwacht, gewartet und verwaltet werden.
Notfall-Wiederherstellungs-Sicht	Diese Sicht beschreibt, wie die Integrationsarchitektur in Notfällen umstrukturiert werden kann, um Verfügbarkeit essenzieller Leistungen zu gewährleisten.
Entwicklungs-Sicht	Diese Sicht beschreibt, wie die Komponenten der Integrationsarchitektur entwickelt werden.

Tabelle 2-7: Sichten zur Beschreibung von Integrationsarchitekturen gemäß Ruh, Maginnes und Brown³⁴⁹

▪ Besonderheiten der Implementierung

Die im Zusammenhang mit den vorangehend behandelten Teilaufgaben aufgezeigten Besonderheiten wirken sich auch auf die Implementierung aus. Entgegen der traditionellen Sichtweise der Softwareentwicklung ist diese Teilaufgabe nicht mehr allein durch die Programmierung neuer Softwarekomponenten geprägt. Es gilt die ausgewählten Integrationsprodukte zu installieren, sie zu konfigurieren und sie in die Integrationslösung einzubinden. Die konzipierten Eingriffe in die Anwendungssysteme zur Modifizierung oder Neuschaffung von externen Schnittstellen werden umgesetzt. Die Komponenten der Integrationslösung werden mittels Adaptern (im Sinne des erläuterten Gestaltungsmusters) und sonstigen „glue-code“ verknüpft.³⁵⁰ Die Realisierung einer Integ-

³⁴⁹ Vgl. Ruh, Maginnis, Brown /EAI/ 169.

³⁵⁰ Vgl. Edwards, Newing /Application Integration/ 203; Lam, Shankararaman /Enterprise Integration/ 47; Schmietendorf u. a. /EAI/ 80.

rationslösung ist zudem durch den Einsatz spezieller Entwicklungswerkzeuge geprägt.³⁵¹

Schließlich empfehlen Edwards und Newing ebenso wie Roch eine dedizierte Entwicklungsumgebung einzurichten.³⁵² Der Gegenstand der Anwendungsintegration ist jedoch die Verknüpfung von bestehenden Anwendungssystemen, die im laufenden Geschäftsbetrieb der Unternehmen eingesetzt und i. d. R. räumlich verteilt betrieben werden. Die Einrichtung einer geeigneten Entwicklungsumgebung dürfte sich daher oftmals als problematisch erweisen.

▪ **Besonderheiten der Überführung in die Nutzung**

Diese Teilaufgabe wird in den angeführten Beiträgen kaum näher für die Anwendungsintegration erläutert. So werden im Wesentlichen die Abnahme der realisierten Integrationslösung durch die Kunden, ihre schrittweise Inbetriebnahme, bei Gewährleistung der Wiederherstellbarkeit des ursprünglichen Zustands für den Fall technischer Probleme sowie die Schulung der Benutzer angesprochen.³⁵³ Einige Autoren weisen darauf hin, dass bei der Anwendungsintegration eine wesentliche Herausforderung darin besteht, die Integrationslösung in die Produktivumgebung zu überführen und dort die abschließende Qualitätssicherung durchzuführen, ohne dass dabei die Unterstützung der laufenden Geschäftsprozesse durch die Anwendungssysteme gefährdet wird.³⁵⁴ Darüber hinaus wird auf die etablierten Konzepte der Softwareentwicklung verwiesen.³⁵⁵

▪ **Besonderheiten der Unterstützungsaufgaben**

Besonderheiten der Unterstützungsaufgaben werden in den angeführten Beiträgen kaum thematisiert. Lediglich die Bedeutung und die Schwierigkeit der dynamischen Qualitätssicherung, d. h. des Testens der Integrationslösungen werden von verschiedenen Autoren betont.³⁵⁶ Beispielsweise empfehlen Lam und Shankararaman die Einrichtung

³⁵¹ Vgl. Linthicum /Application Integration/ 244-257, 376; Jung /Integrationsorientiertes Reengineering/ 23f.

³⁵² Vgl. Edwards, Newing /Application Integration/ 203; Roch /Methodology for EAI – Part II/ 16.

³⁵³ Vgl. Lam, Shankararaman /Enterprise Integration/ 48; Edwards, Newing /Application Integration/ 203; Roch /Methodology for EAI – Part II/ 16.

³⁵⁴ Vgl. Edwards, Newing /Application Integration/ 203; Linthicum /Application Integration/ 388.

³⁵⁵ Vgl. Ruh, Maginnis, Brown /EAI/ 175.

³⁵⁶ Vgl. beispielsweise Linthicum /Application Integration/ 388.

einer dedizierten Testumgebung.³⁵⁷ Dieses Vorhaben dürfte jedoch ebenso wie die Schaffung einer geeigneten Entwicklungsumgebung besondere Herausforderungen bergen.

▪ **Besonderheiten der organisatorischen Gestaltungsmaßnahmen**

Verschiedene Autoren weisen darauf hin, dass Integrationsvorhaben i. d. R. die Zusammenarbeit verschiedener Unternehmensbereiche oder gar verschiedener Unternehmen erfordern, deren Anwendungssysteme verknüpft werden sollen. Darüber hinaus wird angeführt, dass diese beteiligten Parteien oftmals erst wenige Erfahrungen mit der Anwendungsintegration besitzen. Die Autoren fordern daher die Etablierung von zentralen organisatorischen Einheiten, die grundlegende Richtlinien und Vorgaben für die Gestaltung und Umsetzung von Integrationslösungen erarbeiten.³⁵⁸

Yee und Apte weisen darauf hin, dass die Durchführung eines Integrationsvorhabens besondere Ansprüche an die Kenntnisse und Fähigkeiten der Beteiligten stellt. Darauf aufbauend führen sie verschiedene Rollen für Projektbeteiligte an, die aus ihrer Sicht von besonderer Bedeutung für Integrationsvorhaben sind.³⁵⁹ Die Erläuterung der Rollen erfolgt jedoch nur wenig differenziert (siehe Tabelle 2-8).

³⁵⁷ Vgl. Lam, Shankararaman /Enterprise Integration/ 47.

³⁵⁸ In diesem Sinne fordert Roch die Einrichtung eines Integration Competency Center (vgl. Roch /Methodology for EAI – Part I/ 24f.). Ruh, Maginnis und Brown sprechen ihrerseits von der Einrichtung eines Enterprise Architecture Steering Committee oder eines Security Steering Committee (vgl. Ruh, Maginnis, Brown /EAI/ 178).

³⁵⁹ Vgl. Yee, Apte /e-Business Enterprise/ 212f. Rollen basieren auf abgegrenzten Aufgabenkomplexen und beschreiben geforderte Kenntnisse und Fähigkeiten, die ein Rolleninhaber zur Erfüllung der damit verbundenen Tätigkeiten besitzen muss. Ein Mitarbeiter kann mehrere der unterschiedenen Rollen innehaben. Auf der anderen Seite kann eine Rolle von mehreren Mitarbeitern wahrgenommen werden (vgl. Midderhoff /V-Modell/ 68).

Rolle	Erläuterung
Projektförderer (executive sponsor)	Der Inhaber dieser Rolle ist dafür zuständig, das Integrationsvorhaben gegenüber dem Topmanagement der Unternehmen zu repräsentieren und dessen durchgängige Unterstützung sicherzustellen.
Projektleiter (project director)	Der Inhaber dieser Rolle ist für die Planung und Steuerung des Integrationsvorhabens und damit auch für dessen Erfolg verantwortlich.
Chefarchitekt (chief architect)	Der Inhaber dieser Rolle ist für den Entwurf der Integrationslösung und dabei insbesondere für die Gestaltung der Integrationsarchitektur verantwortlich. Dafür benötigt er vor allem fundierte technische Kenntnisse und Fähigkeiten.
Integrationsanalyst (integration analyst)	Diese Rolle wird speziell für Integrationsvorhaben gefordert. Der Inhaber ist insbesondere für die konzeptionelle Modellierung im Rahmen der Analyse des Integrationsproblems und des Entwurfs der Integrationslösung verantwortlich.
Geschäftsfeldanalyst (business domain analyst)	Der Inhaber dieser Rolle ist dafür verantwortlich, die fachlichen Aspekte des Integrationsproblems zu verstehen und die daraus resultierenden Bedürfnisse gegenüber der Integrationslösung zu vertreten.

Tabelle 2-8: Beispiele für Rollen in Integrationsvorhaben nach Yee und Apte³⁶⁰

▪ Zusammenfassende Charakterisierung der Integrationsvorhaben

Aufgrund der bisherigen Erläuterungen wird deutlich, dass sich Integrationsvorhaben durch viele Besonderheiten auszeichnen. Ihnen liegen charakteristische Problemstellungen zugrunde, denen wiederum charakteristische Problemlösungen entgegengestellt werden. Aus der Autonomie, der Verteilung und der Heterogenität der Anwendungssysteme resultieren spezifische Herausforderungen an die beteiligten Personen, die nicht nur technische Mittel, sondern auch organisatorische Maßnahmen erfordern. Dabei vereinen sich in den Integrationsvorhaben Merkmale aus verschiedenen Teildisziplinen der Softwareentwicklung, wie beispielsweise der Neuentwicklung von Softwaresystemen, der Einführung und Anpassung von Standardsoftware sowie dem Reverse Engineering von Altsystemen. Damit begründen sie einen neuartigen Typ von Softwareentwicklungsvorhaben.

³⁶⁰ Vgl. Yee, Apte /e-Business Enterprise/ 212f.

2.3.3 Erfolgsfaktoren und Risiken für Integrationsvorhaben

Gemäß dem dieser Arbeit zugrunde liegenden Verständnis wird ein Integrationsvorhaben durchgeführt, um mit der entwickelten und bereitgestellten Integrationslösung ein bestimmtes Integrationsproblem zu beheben. Unter einem Integrationsproblem wird ein Integrationszustand von Anwendungssystemen verstanden, der als strukturell oder qualitativ unzureichend angesehen wird, einen spezifischen Integrationsbedarf zu erfüllen. Demnach kann der Erfolg eines Integrationsvorhabens vereinfachend anhand von zwei Dimensionen bestimmt werden:

1. Wird mit der Integrationslösung ein Integrationszustand der Anwendungssysteme realisiert, der strukturell und qualitativ geeignet ist, den Integrationsbedarf zu erfüllen (Effektivität des Integrationsvorhabens)?
2. Wurde die Entwicklung und Bereitstellung der Integrationslösung mit einem angemessenen bzw. dem vorgesehenen personellen, zeitlichen und finanziellen Ressourceneinsatz durchgeführt (Effizienz des Integrationsvorhabens)?

Erste Autoren widmen sich den Fragen, welche Faktoren von Bedeutung sind, um ein Integrationsvorhaben erfolgreich abzuschließen (Erfolgsfaktoren), und welche Faktoren auf der anderen Seite eine Bedrohung für den Erfolg eines Integrationsvorhabens darstellen (Risiken).³⁶¹ In diesem Zusammenhang wurden zwei empirische Untersuchungen durchgeführt, die über Einzelfallbetrachtungen hinausgehen. Sie werden im Weiteren vorgestellt werden.

2.3.3.1 Einflussfaktoren auf den Erfolg der Anwendungsintegration

Klesse, Wortmann und Schelp beschreiben die Ergebnisse ihrer Studie, in der „auf der Basis sachlogischer Überlegungen und praktischer Erfahrung“³⁶² ein Kausalmodell aufgestellt und aufgrund quantitativer Daten überprüft wurde und von den Autoren als weitestgehend bestätigt angesehen wird.³⁶³

³⁶¹ Vgl. beispielsweise Klesse, Wortmann, Schelp /Applikationsintegration/; Lam /Enterprise Integration/; Lam /Success Factors/; Schwinn, Winter /Enterprise Application Integration/.

³⁶² Klesse, Wortmann, Schelp /Applikationsintegration/ 260.

³⁶³ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/.

Die der Studie zugrunde liegende Perspektive auf die Anwendungsintegration unterscheidet sich jedoch in drei Punkten von dem Fokus dieser Arbeit:³⁶⁴ Die Autoren betrachten den Erfolg der Anwendungsintegration aus der Sicht der Unternehmen, deren Anwendungssysteme integriert werden. Dabei beziehen sie im Unterschied zur vorliegenden Arbeit auch Gestaltungsmaßnahmen auf den Ebenen der Strategie und der Geschäftsprozesse ein. Der zweite Unterschied besteht in der Beschränkung der Studie auf die innerbetriebliche Anwendungsintegration, während diese Arbeit auch die zwischenbetriebliche Anwendungsintegration in die Betrachtung einschließt. Schließlich bezieht sich die Studie nicht auf den Erfolg abgegrenzter Integrationsvorhaben, sondern sie untersucht den Gesamterfolg aller durchgeführten Integrationsmaßnahmen in den Unternehmen. Die in der Studie betrachteten Erfolgsfaktoren erscheinen trotz dieser Unterschiede für diese Arbeit von Bedeutung, da sie den Kontext beschreiben, in dem einzelne Integrationsvorhaben durchgeführt werden.

Bei der Konzeption des Kausalmodells berufen sich die Autoren auf verschiedene vorangehende Untersuchungen und Fachbeiträge zur Anwendungsintegration.³⁶⁵ Es definiert verschiedene Faktoren, deren Ausprägungen den Erfolg der Anwendungsintegration determinieren sollen. In diesem Sinne werden die Faktoren als Erfolgsvoraussetzungen bezeichnet.

Sowohl die Erfolgsvoraussetzungen als auch der Erfolg werden von den Autoren als latente Größen, d. h. als nicht unmittelbar messbar angesehen. Darüber hinaus werden sie von den Autoren als komplex und multidimensional charakterisiert. Im Zuge ihrer Operationalisierung werden sie daher jeweils in mehrere Komponenten aufgeteilt.³⁶⁶ Die Erfolgsvoraussetzungen ebenso wie der Erfolg stellen damit abstrakte und zusammengesetzte Attribute dar, deren Ausprägungen durch die Beurteilungen aller zugehörigen Komponenten bestimmt werden.³⁶⁷

Der Erfolg der Anwendungsintegration wird auf diese Weise durch fünf Komponenten gebildet: die Kosteneffizienz der Anwendungsintegration, die Qualität der Geschäfts-

³⁶⁴ Vgl. im Folgenden Klesse, Wortmann, Schelp /Applikationsintegration/ 259f.

³⁶⁵ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ 260f.

³⁶⁶ Dieses Verständnis wird beispielsweise anhand der folgenden Formulierung deutlich: "Als Indikatoren für ein effektives Architekturmanagement können die Umsetzungsgrade dessen wesentlicher Bestandteile dienen." (Klesse, Wortmann, Schelp /Applikationsintegration/ 261).

³⁶⁷ Vgl. Rossiter /Scale Development/ 314f.

prozessunterstützung, die Einhaltung zeitlicher Projektziele, die unternehmerische Flexibilität und die Zufriedenheit der Anwender mit den Integrationslösungen.³⁶⁸

Dem Erfolg werden acht Erfolgsvoraussetzungen gegenüber gestellt, die der Ebene der Strategie, der Ebene der Geschäftsprozesse und der Ebene der Anwendungssysteme zugeordnet werden.³⁶⁹ Als zentrale Erfolgsvoraussetzung auf der Strategiebene wird ein funktionierendes Architekturmanagement angesehen, welches die in der IT-Strategie formulierten Ziele für die Anwendungsintegration konkretisiert und operationalisiert.³⁷⁰

Der Ebene der Geschäftsprozesse werden drei Erfolgsvoraussetzungen zugeordnet. Dies ist zum einen das Business-IT-Alignment, welches die Abstimmung zwischen IT- und Geschäftsstrategie sowie die Gestaltung einer effizienten prozessualen Schnittstelle zwischen IT und Fachbereich umfasst. Zum anderen wird ein hoher Reifegrad der Organisation der Integration als erforderlich angenommen, wozu definierte Prozesse für das Architekturmanagement, die Anwendungsintegration sowie die Entwicklung und den Betrieb der Anwendungssysteme gezählt werden. „Diese einzelnen Prozesse sollten aufeinander abgestimmt und miteinander integriert sein, um ihre Wirksamkeit zu gewährleisten und Ineffizienzen zu vermeiden.“³⁷¹ Schließlich wird eine methodische Unterstützung der Anwendungsintegration als Erfolgsvoraussetzung genannt.³⁷²

Auf der Ebene der Anwendungssysteme werden darüber hinaus vier Erfolgsvoraussetzungen angenommen: die Strukturierung der Anwendungslandschaft nach den Grundsätzen der serviceorientierten Architektur, die Konsolidierung der Anwendungslandschaft, die Standardisierung der Integrationsinfrastrukturen sowie die technische Qualität der Integrationsinfrastrukturen.

Die von den Autoren vorgenommene Analyse der Erfolgsvoraussetzungen in ihre Komponenten wird in Tabelle 2-9 wiedergegeben.

³⁶⁸ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ 260.

³⁶⁹ Die Ebene der Anwendungssysteme wird von den Autoren als Systemebene bezeichnet und in die Anwendungssysteme sowie die Integrationsinfrastruktur differenziert.

³⁷⁰ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ 261.

³⁷¹ Klesse, Wortmann, Schelp /Applikationsintegration/ 262.

³⁷² Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ 262.

Erfolgsvoraussetzung	Komponente
Architekturmanagement	Eine Integrationsstrategie ist formuliert und unternehmensweit durchgesetzt.
	Zur Entwicklung/Planung der Zielarchitekturen und zur Erfassung des Ist-Zustandes werden Modelle erstellt.
	Um die konzipierten Zielarchitekturen zu realisieren, werden Umsetzungs- und Transformationsregeln formuliert und durchgesetzt, wie Architekturprinzipien und -leitlinien.
	Zur Steuerung und Kontrolle der Architekturumsetzung und zur Gewährleistung der Lenkbarkeit der Architektur werden Kennzahlensysteme eingesetzt.
Business-IT-Alignment	Es besteht eine hohe Kooperationsfähigkeit zwischen IT und Fachabteilungen.
	Die Geschäftsprozesse sind bekannt und dokumentiert.
	Die Lösungsfindung in der IT erfolgt geschäftsorientiert.
Reifegrad der Organisation der Integration	Die Ablauforganisation ist dokumentiert.
	Es erfolgt eine klare Zuweisung der Verantwortlichkeiten zu den einzelnen Aufgabenträgern der Anwendungsintegration.
	Es ist ein reibungsloses, integriertes Zusammenspiel der Prozesse Anwendungsintegration, Architekturmanagement, Anwendungsentwicklung und -betrieb sowie Entwicklung und Betrieb der Integrationsinfrastruktur gewährleistet.
	Die am Integrationsprozess beteiligten Mitarbeiter verfügen über entsprechendes Wissen und Erfahrung.
	Eine durchgängige Qualitätssicherung ist gewährleistet.
Methodische Unterstützung	Methoden zur Anwendungsintegration stehen zur Verfügung und werden angewandt.
	Die Gestaltung der Integrationsarchitektur erfolgt aufgrund von Gestaltungsmustern.
Serviceorientierte Architektur	Die Schichten der Benutzerschnittstelle, der Anwendungsfunktionalität und der Datenverwaltung sind getrennt.
	Die Funktionalität ist in unabhängige Module und Dienste untergliedert.
	Schnittstellen sind explizit spezifiziert und dokumentiert.
	Während die Funktionalität in den einzelnen Modulen und Diensten realisiert ist, steuert und verwaltet ein zentrales, autonomes Workflowmanagementsystem den Kontrollfluss.
	Es wird eine lose Kopplung unter den Anwendungssystemen sichergestellt.
Konsolidierung der Anwendungslandschaft	Die eingesetzten Anwendungssysteme sind weitgehend Standardsoftwaresysteme.
	Die Gesamtanzahl der Anwendungssysteme wird reduziert.
	Die Gesamtanzahl unterschiedlicher Systemplattformen wird reduziert.
Standardisierung der Integrationsinfrastruktur	Integrationsfunktionalität wird in eine dedizierte Integrationsinfrastruktur ausgelagert.
	In der Integrationsinfrastruktur werden durchgehend Standards verwendet.
	Es wird eine geringe Anzahl unterschiedlicher Integrationsprodukte eingesetzt.
	Integrationsprodukte werden unternehmensweit eingesetzt.
Technische Qualität	Es werden performante, hochverfügbare und skalierbare Integrationsprodukte eingesetzt.
	Vorgefertigte Adapter stehen zur Verfügung und werden eingesetzt.

Tabelle 2-9: Komponenten der Erfolgsvoraussetzungen der Anwendungsintegration gemäß Klesse, Wortmann und Schelp³⁷³

³⁷³ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ 261-263.

Die Datenerhebung erfolgte anhand eines schriftlichen Fragebogens unter den Teilnehmern des 14. St. Galler Anwenderforums. Der Umsetzungs- bzw. Erreichungsgrad der Komponenten der Erfolgsvoraussetzungen und des Erfolgs der Anwendungsintegration wurde durch die Studienteilnehmer auf vierstufigen Skalen beurteilt. Fünfundneunzig der auf diese Weise erhobenen Datensätze wurden in die Kausalanalyse einbezogen, aufgrund derer unterschiedlich starke Einflüsse der genannten Erfolgsvoraussetzungen auf den Erfolg der Anwendungsintegration festgestellt wurden.³⁷⁴ Die angewandten Analysemethoden ebenso wie die Interpretation der Ergebnisse werden den gemessenen Attributen jedoch nicht gerecht. Beispielsweise verweisen die Autoren darauf, dass ein Zusammenhang zwischen der Zufriedenheit der Anwender und dem Erfolg der Anwendungsintegration statistisch nicht nachgewiesen werden konnte. Dies widerspricht der früheren Aussage, dass der Erfolg der Anwendungsintegration aus dem Erreichungsgrad der unterschiedenen Komponenten resultiert.³⁷⁵ Auch die Berechnungen, welcher Anteil der Varianzen der einzelnen Komponenten durch die jeweiligen Attribute erklärt werden, erscheinen nicht als sinnvoll.³⁷⁶

2.3.3.2 Risiken für Integrationsvorhaben

Lam stellt die Ergebnisse einer qualitativen Studie zu den Risiken für Integrationsvorhaben vor.³⁷⁷ Im Rahmen eines gemeinsamen Workshops wurden die Ansichten von 16 erfahrenen Praktikern zu den maßgeblichen Risiken für Integrationsvorhaben sowie ihre Empfehlungen für einen angemessenen Umgang mit den Risiken gesammelt und diskutiert.³⁷⁸

Die Diskussionsgrundlage bildete ein Modell von Integrationsvorhaben, das die vier Phasen Strategie, Planung, Implementierung und Überführung in die Nutzung (Roll-out) unterscheidet. Diesen Phasen sind jeweils abstrakt beschriebene Tätigkeiten zugeordnet,

³⁷⁴ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ 263-265.

³⁷⁵ „Der Erfolg der Applikationsintegration bemisst sich also daran, ob das Unternehmen die selbstgesetzten Ziele bezüglich der Applikationsintegration erreicht hat.“ (Klesse, Wortmann, Schelp /Applikationsintegration/ 260)

³⁷⁶ Vgl. Rossiter /Scale Development/ 314f. Das kritikwürdige Verständnis der Autoren für die Untersuchung formativer Attribute wird bereits darin ersichtlich, dass sie die einzelnen Komponenten wiederholt als Indikatoren bezeichnen, obwohl diese gemäß Rossiter als eigenständige Attribute anzusehen sind (vgl. Rossiter /Scale Development/ 314). Siehe diesbezüglich auch Kapitel 5.2.2.2.

³⁷⁷ Vgl. Lam /Enterprise Integration/.

³⁷⁸ Vgl. Lam /Enterprise Integration/ 298f.

die sich auf den Geschäftsbetrieb und die organisatorische Gestaltung der Unternehmen sowie die Entwicklung der Integrationslösung beziehen.³⁷⁹

Insgesamt wurden in der Diskussion 33 Risikobereiche abgegrenzt, die den unterschiedlichen Phasen eines Integrationsvorhabens zugeordnet wurden. Diese Ergebnisse sind in Tabelle 2-10 dargestellt.³⁸⁰

Phase der Integrationsvorhaben	Risikobereich
Strategie	Unklare geschäftliche Vision für die Integration der Anwendungssysteme.
	Mangelnde Einbeziehung und Unterstützung der geschäftlichen stakeholder des Integrationsvorhabens.
	Mangelhaft verfasster business case für das Integrationsvorhaben mit einem unklaren return on investment (ROI).
	Mangelhafte Untersuchung der organisatorischen Auswirkungen der Integrationslösung.
Planung	Die geschäftlichen Anforderungen an die Integrationslösung sind nicht vollständig verstanden.
	Unzureichende zeitliche Planung und Synchronisierung des Integrationsvorhabens.
	Unrealistische Schätzung des Entwicklungsaufwands.
	Im Projektverlauf zunehmende Anforderungen an die Integrationslösung („scope creep“).
	Isolierte Teilprojekte infolge einer fehlenden Gesamtkonzeption der Integrationsarchitektur aufgrund von Standards.
	Mangelnde Fähigkeiten und Erfahrungen führen zu falschen Entwicklungsentscheidungen.
	Mentalität der vorgefertigten („out-of-the-box“) Integrationslösung.
	Unterlassenes Geschäftsprozessmanagement und unmittelbares Beginnen mit der technischen Entwicklungsarbeit.
	Konzentration auf einen einzigen Lieferanten für Systemtechnik, anstelle eines „best-of-breed“-Ansatzes.
	Unzureichend definierte organisatorische Strukturen und Rollen.
Implementierung (Fortsetzung folgende Seite)	Mangel eines vereinbarten und wohldefinierten Modells der Geschäftsprozesse.
	Semantische Konflikte der Daten und Mangel eines vereinbarten umfassenden Datenmodells.
	Verwendung von proprietärer Technologie anstelle von offenen Standards.
	Performanzprobleme der Integrationslösung.
	Gefährliche Entwicklung hin zu einer Punkt-zu-Punkt Integration der Anwendungssysteme.
	Verteilung von Geschäftslogik über die Integrationsarchitektur.

³⁷⁹ Vgl. Lam /Enterprise Integration/ 300f.

³⁸⁰ Vgl. Lam /Enterprise Integration/ 303-310.

Phase der Integrationsvorhaben	Risikobereich
Implementierung (Fortsetzung)	Einsatz von unreifen Technologien und Integrationsprodukten, die nicht für die Entwicklung geschäftskritischer Integrationslösungen geeignet sind.
	Beschränkte externe Schnittstellen der Anwendungssysteme und Integrationsprodukte.
	Verletzung der Integrität der Datenbestände infolge der Anbindung der Anwendungssysteme auf der Schicht der Datenverwaltung.
	Fehlende Konzeption eines durchgängigen Sicherheitskonzepts für die Integrationslösung.
	Komplizierte Eingriffe in Altsysteme.
	Unzureichende Planung oder Kommunikation der Inbetriebnahme von Geschäftsabläufen, die von der Integrationslösung abhängen.
	Kein vorbereitetes Change Management.
Überführung in die Nutzung (Roll-out)	Zu starke Fokussierung des Release-Plans auf Integrationsbereiche von geringem Nutzen.
	Mangel an geeigneten Plänen für die Inbetriebnahme und Migration.
	Upgrades von Softwarekomponenten verursachen ein Fehlverhalten von Adaptern oder Integrationsprodukten.
	Fehlende Flexibilität für die Unterstützung neuer geschäftlicher Anforderungen an die Integrationslösung.
	Mangelnde Evaluation der abgeschlossenen Anwendungsintegration.
	Mangelnde Governance-Strukturen für die Anwendungsintegration.

Tabelle 2-10: Risikobereiche von Integrationsvorhaben gemäß Lam³⁸¹

2.4 Entwicklungsaufwand und die ihn beeinflussenden Faktoren

Softwareentwicklungs- bzw. Integrationsvorhaben erfordern die Erbringung personeller Arbeitsleistung, d. h. sie sind mit Aufwand verbunden. Im Weiteren werden ausgewählte Erkenntnisse zu den ihn beeinflussenden Faktoren aus verschiedenen Teildisziplinen der Softwareentwicklung vorgestellt. Einleitend werden dazu einige grundlegenden Überlegungen zum Entwicklungsaufwand und zur Untersuchung der ihn beeinflussenden Faktoren angestellt.

2.4.1 Grundlegende Überlegungen zum Entwicklungsaufwand

Die vorliegende Arbeit widmet sich der Forschungsfrage, welche Faktoren in die Aufwandschätzung von Integrationsvorhaben einbezogen werden müssen, um zuverlässige Schätzergebnisse zu gewährleisten. Um diese Frage weiter verfolgen zu können, erscheint es erforderlich, ein grundlegendes Verständnis vom Entwicklungsaufwand und

³⁸¹ Vgl. Lam /Enterprise Integration/ 303-310.

den ihn beeinflussenden Faktoren zu gewinnen. Dabei können die weiteren Überlegungen als allgemeingültig für die Softwareentwicklung angesehen werden. Sie sind damit auch für Integrationsvorhaben relevant.

Entwicklungsaufwand wurde definiert als mengenmäßiger Verbrauch von personeller Leistung für die Entwicklung und Bereitstellung eines Softwaresystems, ausgedrückt als die Summe der Arbeitszeit, die Personen für die Erfüllung der mit dem Entwicklungsvorhaben verbundenen Aufgaben einsetzen. Aufbauend auf diesem Verständnis können zwei dem Entwicklungsaufwand zugrunde liegende Größen unterschieden werden: die zu bewältigende Arbeitslast und die Produktivität der Mitarbeiter, d. h. welche Arbeit erbracht werden muss und wie viel Zeit die Mitarbeiter dafür benötigen.

Die Arbeitslast resultiert insbesondere aus den zu erfüllenden Entwicklungs- und Unterstützungsaufgaben. Somit kann der Entwicklungsaufwand durch die Betrachtung einzelner Teilaufgaben der Entwicklungsvorhaben oder der mit ihnen verbundenen Arbeitsergebnisse untergliedert werden.

Die Produktivität der Mitarbeiter bezeichnet die Menge an Arbeitslast, die in einer bestimmten Zeit bewältigt wird.³⁸² Wie erläutert werden Entwicklungsvorhaben oftmals arbeitsteilig, d. h. von einer Mehrzahl von Personen erbracht. Es resultiert daraus ein Netzwerk von Leistungsbeiträgen verschiedener Personen.³⁸³ Die Produktivität der Mitarbeiter wird demnach durch ihre individuellen Leistungen sowie die aus ihrer Zusammenarbeit resultierenden Effekte geprägt.

Ein Einflussfaktor auf den Entwicklungsaufwand kann demnach als eine Größe definiert werden, die verstärkend oder vermindern auf die Arbeitslast oder die Produktivität im Rahmen eines Entwicklungsvorhabens wirkt.

Diese umfassende Definition spiegelt wider, dass von einer geradezu unfassbaren Menge von Einflussfaktoren auszugehen ist. Sämtliche Größen, die auf die zu bewältigende Arbeitslast oder die Produktivität der Mitarbeiter wirken, beeinflussen damit auch den Aufwand. Dabei muss angenommen werden, dass einzelne Einflussfaktoren vielfältige Auswirkungen auf die beiden Komponenten des Entwicklungsaufwands haben können.

³⁸² Vgl. Noth, Kretschmar /Aufwandschätzung/ 18.

³⁸³ Vgl. Witte /Entscheidungsprozesse/ 552-554; Putz-Osterloh /Entscheidungsverhalten/ 585f.

Darüber hinaus können zwischen den einzelnen Einflussfaktoren vielfältige Beziehungen bestehen, sowohl im Hinblick auf ihr Auftreten als auch auf ihre Auswirkungen. Selbst zwischen den beiden Komponenten des Entwicklungsaufwands, der zu bewältigenden Arbeitslast und der Produktivität der Mitarbeiter, konnten Wechselwirkungen festgestellt werden.³⁸⁴ Die Einflussfaktoren auf den Entwicklungsaufwand können demnach als Knotenpunkte in einem komplexen Netzwerk aus Ursachen und Wirkungen verstanden werden. Daraus folgend besitzen sie in der Regel einen nicht-linearen und vielfach zeitverzögerten Einfluss auf den Aufwand.³⁸⁵

Des Weiteren stellen einzelne Entwicklungsvorhaben in ihrer Gesamtheit einzigartige Konstellationen aufwandsrelevanter Sachverhalte dar. Beispielsweise gibt es viele unterschiedliche Arten von zu entwickelnden Softwaresystemen, die jeweiligen Projektbeteiligten weisen in vielerlei Hinsicht individuelle Merkmale auf, eine Vielzahl unterschiedlicher Methoden ebenso wie eine große Bandbreite von Technologien kommen in der Softwareentwicklung zum Einsatz. Die Ausprägungen einzelner Einflussfaktoren ebenso wie ihre Auswirkungen auf den Aufwand können daher zwischen verschiedenen Entwicklungsvorhaben stark variieren.³⁸⁶

2.4.2 Die kontinuierliche Suche nach den Determinanten des Entwicklungsaufwands

Aus den erläuterten Merkmalen des Entwicklungsaufwands und der auf ihn wirkenden Größen resultieren drei Herausforderungen für die Bemühungen um ein besseres Verständnis der Einflussfaktoren: die Komplexitätsreduktion, die Bestimmung des Gültigkeitsbereichs der Erkenntnisse und das Problem der dynamischen Veränderung der Grundlagen.

Die Einflussfaktoren auf den Entwicklungsaufwand wurden als Knotenpunkte in einem komplexen Netzwerk aus Ursachen und Wirkungen charakterisiert. Es erscheint daher nicht möglich, die Wirkungspfade in diesem Netzwerk zu eindeutigen Ursprüngen zu-

³⁸⁴ Beispielsweise wird dem u. a. auf der gegebenen Arbeitslast basierenden Leistungsdruck große Bedeutung für die Leistungsfähigkeit und Leistungsbereitschaft der Mitarbeiter beigemessen (vgl. Craig, Jassim /Management for IT/ 142).

³⁸⁵ Vgl. Abdel-Hamid, Madnick /Dynamics/ 1428f.; Milling /Komplexe Systeme/ 3.

³⁸⁶ Vgl. Jones /Software Costs/ 25.

rückzuverfolgen. Vielmehr gilt es im Zuge der Komplexitätsreduktion, diejenigen Knotenpunkte zu erfassen und Ausschnitte ihrer Beziehungen zu approximieren, die den Entwicklungsaufwand besonders signifikant beeinflussen.

Ferner wurde darauf hingewiesen, dass einzelne Entwicklungsvorhaben in ihrer Gesamtheit einzigartige Konstellationen aufwandsrelevanter Sachverhalte darstellen. Es gilt daher, in sich hinreichend homogene Arten von Entwicklungsvorhaben abzugrenzen, für die gemeingültige Erkenntnisse zu den Einflussfaktoren gewonnen werden können.³⁸⁷ Anhand welcher Merkmale die Einteilung der Entwicklungsvorhaben erfolgen sollte, kann jedoch nur aufgrund des Verständnisses der Einflussfaktoren abschließend geklärt werden. Die Bestimmung des Gültigkeitsbereichs der Erkenntnisse erweist sich damit als Teufelskreis.

Aufgrund der dynamischen Entwicklung der Grundlagen der Softwareentwicklung, wie beispielsweise der Technologien und Entwicklungsparadigmen, der Einsatzgebiete der Softwaresysteme oder der verfügbaren Mitarbeiter, stellt sich die Aufgabe, die gewonnenen Erkenntnisse stets hinsichtlich ihrer aktuellen Gültigkeit zu prüfen und sie nötigenfalls an die jeweils aktuellen Charakteristika der Entwicklungsvorhaben anzupassen.³⁸⁸ Da dabei die vorangehenden Aufgaben der Identifikation der relevanten Einflussfaktoren und der Analyse der bestehenden Wirkungsbeziehungen sowie die Identifikation verallgemeinerbarer Gesetzmäßigkeiten und die Bestimmung der Grenzen der Übertragbarkeit erneut anfallen, erweist sich die Suche nach den Determinanten des Entwicklungsaufwands als kontinuierliche Aufgabe.

Infolge dieser Herausforderungen wird in der Fachliteratur eine Vielzahl unterschiedlicher Aussagen zu den Einflussfaktoren auf den Entwicklungsaufwand vertreten, deren Relevanz, Übertragbarkeit und Aktualität neben der wissenschaftlichen Qualität der zugrunde liegenden Untersuchungen kontrovers diskutiert werden.³⁸⁹ Es kann daher der Eindruck gewonnen werden, die Aufwandsschätzung sei „mehr eine Kunst als eine Wissenschaft.“³⁹⁰

³⁸⁷ Vgl. Jones /Software Costs/ 25.

³⁸⁸ So wird als generelles Charakteristikum von Modellen genannt, dass sie temporär problemadäquat und zeitabhängig sind (vgl. Dresbach /Modelle/ 89).

³⁸⁹ Vgl. beispielsweise Jones /Software Costs/; Stevenson /Software Engineering Productivity/.

³⁹⁰ „Software cost estimation is more of an art than a science.“ (Pillai, Nair /Estimation/ 485)

3 Stand der Erkenntnis zu den Einflussfaktoren auf den Entwicklungsaufwand

Im vorangehenden Kapitel wurden die Charakteristika der Anwendungsintegration im Hinblick auf die Integrationsprobleme, die Integrationslösungen und die Integrationsvorhaben erläutert. Diesem konzeptionellen Bezugsrahmen wird im Weiteren der Stand der Erkenntnis zu den Einflussfaktoren auf den Entwicklungsaufwand gegenübergestellt. Dabei tritt die der Arbeit zugrunde liegende Problemstellung zutage: es sind bislang keine wissenschaftlichen Ergebnisse bekannt, die sich unmittelbar auf die Anwendungsintegration im Sinne dieser Arbeit beziehen.

In Ermangelung an spezifischen Erkenntnissen werden mehrere Untersuchungen und Verfahren zur Aufwandschätzung aus verschiedenen Teildisziplinen der Softwareentwicklung aufgegriffen. Dabei wird zunächst auf die klassische Softwareentwicklung eingegangen: die Neuentwicklung von Anwendungssystemen. Diesbezüglich wurde über die Jahre ein umfangreicher Komplex von Untersuchungsergebnissen und Verfahren zur Aufwandschätzung aufgebaut, der in dieser Arbeit lediglich in möglichst repräsentativen Ausschnitten erfasst werden kann.³⁹¹ Anschließend werden erste Ergebnisse aus anderen Teildisziplinen der Softwareentwicklung aufgegriffen, die einen engen Bezug zu den geschilderten Aufgabenmerkmalen der Anwendungsintegration aufweisen: die Softwareentwicklung unter Einsatz von Standardsoftwarekomponenten, die Verknüpfung selbständiger Softwaresysteme zu einem umfassenden System (System of Systems, SoS) und schließlich das integrationsorientierte Reengineering von Altsystemen.

Der Betrachtungsschwerpunkt wird im Sinne der verfolgten Forschungsfrage darauf gelegt, welche Größen in verschiedenen Zusammenhängen als relevant angesehen werden. Eine erschöpfende Diskussion der Stärken und Schwächen der vorgestellten Beiträge hinsichtlich der Operationalisierung und Messung der Einflussfaktoren wird daher nicht angestrebt.

³⁹¹ Umfassendere Darstellungen bieten diesbezüglich Jones /Software Costs/ sowie Stevenson /Software Engineering Productivity/.

3.1 Erkenntnisse zur Neuentwicklung von Anwendungssystemen

3.1.1 Der Umfang der zu entwickelnden Software als Arbeitslast

Die Arbeitslast der Softwareentwicklung wird in der Fachliteratur überwiegend auf den Umfang der zu entwickelnden Software zurückgeführt. Dieser wird somit als grundlegender Einflussfaktor auf den Entwicklungsaufwand angesehen, während weitere Aspekte der zu entwickelnden Software oder des durchzuführenden Entwicklungsvorhabens als Einflüsse auf die Produktivität der Mitarbeiter verstanden werden. Welcher Aufwand für die Entwicklung einer Software eines bestimmten Umfangs benötigt wird, hängt demnach davon ab, wie diese weiteren Einflussfaktoren ausgeprägt sind.³⁹²

Bundschuh und Fabry erläutern dieses Verständnis anhand eines Beispiels. Die Entwicklung einer Software wird dabei mit der Durchführung einer Reise verglichen, wobei der Entwicklungsaufwand seine Entsprechung in der erforderlichen Reisezeit findet. Die Bestimmung des Umfangs der zu entwickelnden Software wird mit der Bestimmung der mit der Reise zurückzulegenden Entfernung verglichen. Darauf aufbauend können die weiteren Einflussfaktoren mit der Auswahl des Verkehrsmittels, der Festlegung der durchzuführenden Pausen, der Ortskundigkeit des Reisenden u. ä. verglichen werden.³⁹³

Die Frage, was den Umfang einer zu entwickelnden Software ausmacht, wird auf unterschiedliche Weise beantwortet, beispielsweise mit Bezug auf den Programmcode oder die Funktionalität der Software³⁹⁴

³⁹² Vgl. Jones /Software Costs/ 5f.; Balzert /Software-Technik/ 74; Stevenson /Software Engineering Productivity/ 10-23; Bundschuh, Fabry /Aufwandschätzung/ 21-23.

³⁹³ Vgl. Bundschuh, Fabry /Aufwandschätzung/ 21.

³⁹⁴ Im Weiteren werden die Ansätze vorgestellt, den Umfang eines Softwaresystems anhand der Zeilen des Programmcodes, der Anzahl der logischen Anweisungen oder der Function Points zu erfassen, da sie in Forschung und Praxis den großen Anklang finden. Darüber hinaus wurden in Theorie und Praxis verschiedene andere Vorschläge entwickelt, die ebenfalls den Umfang eines Softwaresystems betreffen. Sie beziehen sich beispielsweise auf die Anzahl der Module des Softwaresystems oder Art und Umfang der Dokumentation der Software (vgl. beispielsweise Stevenson /Software Engineering Productivity/ 17-22).

▪ **Merkmale des Programmcodes als Ausdruck des Umfangs einer Software**

Viele Autoren verstehen die Anzahl der Zeilen des Programmcodes (Lines of Code) oder der Anzahl der im Programmcode enthaltenen logischen Anweisungen (Logical Statements) als Ausdruck des Umfangs einer Software.³⁹⁵ In verschiedenen Studien wurde beispielsweise ein signifikanter Zusammenhang zwischen der Anzahl der Programmzeilen einer entwickelten Software und dem dafür benötigten Entwicklungsaufwand festgestellt.³⁹⁶

Die Allgemeingültigkeit dieser Zusammenhänge wird jedoch vielfach in Frage gestellt. Sowohl die Anzahl der Programmzeilen einer Software ebenso wie die Anzahl der enthaltenen logischen Anweisungen variiert mit der verwendeten Programmiersprache und der Programmierweise. Insbesondere für Entwicklungsvorhaben, bei denen verschiedene Programmiersprachen gleichzeitig eingesetzt werden, kann damit oftmals nicht von einem einheitlichen Einfluss auf den Entwicklungsaufwand ausgegangen werden. Als problematisch wird auch angesehen, wie bestimmte Teile des Programmcodes in die Zählung einbezogen werden sollten. So können Teile des Programmcodes wiederverwendet oder durch Entwicklungswerkzeuge automatisch generiert worden sein, die unverändert gelassen oder modifiziert worden sind. Auf der anderen Seite müssen nicht alle Teile des entwickelten Programmcodes auch in der fertigen Software enthalten sein.³⁹⁷

Die Anzahl der Zeilen des Programmcodes der zu entwickelnden Software ebenso wie die Anzahl der im Programmcode enthaltenen logischen Anweisungen können damit trotz ihrer verbreiteten Anwendung nicht uneingeschränkt als maßgebliche Einflussfaktoren auf den gesamten Entwicklungsaufwand der Software angesehen werden.

³⁹⁵ Vgl. Jones /Software Costs/ 317-319.

³⁹⁶ Vgl. beispielsweise Walston, Felix /Programming Measurement and Estimation/ 62; Jones /Software Costs/ 174-177;

³⁹⁷ Vgl. Jones /Software Costs/ 319f., Stevenson /Software Engineering Productivity/ 10-15; Humphrey, Singpurwalla /Predicting Software Productivity/ 196. Aufgrund dieser Kritikpunkte wird das Verständnis der für den Entwicklungsaufwand maßgeblichen Charakteristika des Programmcodes oftmals präzisiert. Beispielsweise bezieht sich Stevenson auf die Anzahl der vollständig fehlerfreien und dokumentierten logischen Anweisungen zur Bestimmung des Umfangs einer Software (vgl. Stevenson /Software Engineering Productivity/ 17). Dennoch können damit nicht alle der aufgeführten Kritikpunkte ausgeräumt werden.

▪ Die Funktionen einer Software als Ausdruck ihres Umfangs

Als Reaktion auf die genannten Kritikpunkte wurden verschiedene Ansätze entwickelt, den Umfang durch die Menge und Art der Funktionen der Software auszudrücken.³⁹⁸ Auf diese Weise sollte ein Verständnis des Umfangs einer Software begründet werden, das unabhängig von den für die Entwicklung eingesetzten Technologien ist.³⁹⁹ Insbesondere die von Albrecht veröffentlichte Metrik der Function Points wird für diese Ansätze als grundlegend angesehen und hat große Verbreitung erfahren.⁴⁰⁰

Das Konzept der Function Points wird im Weiteren anhand der von der International Function Point Users Group (IFPUG) veröffentlichten und als internationaler Standard anerkannten Regelungen dargestellt.⁴⁰¹ Dabei steht wiederum nicht das Vorgehen zur Bestimmung des Umfangs, sondern vielmehr das der Metrik zugrunde liegende Verständnis der Einflussfaktoren auf den Entwicklungsaufwand im Vordergrund.

Es werden fünf Kategorien von Funktionen eines Softwaresystems unterschieden: Externe Eingaben (External Inputs), Externe Ausgaben (External Outputs), Externe Abfragen (External Inquiries), Interne Datenbestände (Internal Logical Files) und Externe Schnittstellen-Datenbestände (External Interface Files), d. h. Daten, die außerhalb der Grenzen des Softwaresystems und somit von anderen Softwaresystemen gepflegt werden. Jede dieser Kategorien wird weiterhin hinsichtlich des Komplexitätsgrads in einfache, mittlere und komplexe Funktionen unterteilt.⁴⁰² Eine Funktion einer Software wird dann je nach Kategorie und Klassifizierung einer definierten Menge von sog. ungewich-

³⁹⁸ Die betrachteten Funktionen einer Software stellen das Ergebnis von Entwurfsentscheidungen dar. Es erscheint daher als unpräzise, wenn verschiedene Autoren in diesem Zusammenhang von Anforderungen des Kunden sprechen (vgl. z. B. Balzert /Software-Technik/ 83; Bundschuh, Fabry /Aufwandschätzung/ 179).

³⁹⁹ Vgl. Albrecht /Productivity/ 84.

⁴⁰⁰ Vgl. Albrecht /Productivity/ 83-86. Ein ähnlicher Vorschlag zum Bestimmen des Umfangs einer Software aufgrund der funktionsbezogenen Anforderungen, die sog. Function Bang Metrik, wurde kurz darauf von DeMarco veröffentlicht (vgl. DeMarco /Controlling/ 49-91). In einem Experiment von Rask, Laamanen und Lyytinen konnte keine einheitliche Korrelation zwischen dem Ausdruck des Umfangs desselben Softwaresystems in Function Points und in Function Bangs festgestellt werden. In manchen Situationen führten die beiden Ansätze zu stark unterschiedlichen Aussagen zum Umfang des Softwaresystems (vgl. Rask, Laamanen, Lyytinen /Function Point and Function Bang/ 661-670).

⁴⁰¹ Vgl. IFPUG /Counting Practices Manual/, dargestellt anhand von Bundschuh, Fabry /Aufwandschätzung/ 179-268.

⁴⁰² In welche dieser Gruppen eine Funktion zu klassifizieren ist, wird zum einen anhand der Anzahl der betroffenen Datenelemente bestimmt. Für die Internen Datenbestände und Externen Schnittstellen-Datenbestände wird darüber hinaus die Anzahl der Untergruppen der Datenelemente einbezogen, während für die üblichen Kategorien die Anzahl der bei der Abarbeitung der Funktionen benötigten internen oder externen Datenbestände einbezogen werden.

teten Function Points gleichgesetzt (vgl. Tabelle 3-1). Somit werden Anzahl, Kategorie und Komplexität der Funktionen einer Software als grundlegende Einflussfaktoren auf den Entwicklungsaufwand mit spezifischer Wirkungsstärke angesehen.

Bei der Abgrenzung der betrachteten Funktionen soll die Betrachtungsperspektive des Benutzers eingenommen werden. Diese Einflussfaktoren stehen damit für konzeptionelle, nicht für technische Charakteristika der zu entwickelnden Software.⁴⁰³

Ferner ist zu berücksichtigen, ob es sich bei einem Entwicklungsvorhaben um die Neuentwicklung oder die Weiterentwicklung eines Softwaresystems handelt. Im ersten Fall werden lediglich die neu entwickelten Funktionen erfasst. Im zweiten Fall werden hingegen die hinzugefügten, die geänderten sowie die entfernten Funktionen einbezogen.⁴⁰⁴

Kategorie	Klassifizierung	Function Points je entsprechender Funktion
Externe Eingaben (External Inputs)	einfach	3
	mittel	4
	komplex	6
Externe Ausgaben (External Outputs)	einfach	3
	mittel	4
	komplex	6
Externe Abfragen (External Inquiries)	einfach	4
	mittel	5
	komplex	7
Interne Datenbestände (Internal Logical Files)	einfach	7
	mittel	10
	komplex	15
Externe Schnittstellen-Datenbestände (External Interface Files)	einfach	5
	mittel	7
	komplex	10

Tabelle 3-1: Kategorien und Klassifizierung von Funktionen zum Zählen von Function Points⁴⁰⁵

Das der Function Point Metrik zugrunde liegende Verständnis des Umfangs einer Software wird jedoch ebenfalls kritisiert. Die unterschiedenen Kategorien von Funktionen scheinen nicht für alle Arten von Softwaresystemen gleichermaßen angemessen. Die

⁴⁰³ Vgl. Bundschuh, Fabry /Aufwandschätzung/ 190-193. Das Verständnis dieser Einflussfaktoren wird in den detaillierten Regelungen zum Zählen der Funktionen präzisiert. Eine Darstellung dieser Zählregelungen erfolgt beispielsweise in Bundschuh, Fabry /Aufwandschätzung/ 241-252.

⁴⁰⁴ Vgl. Bundschuh, Fabry /Aufwandschätzung/ 188f.

⁴⁰⁵ Vgl. Bundschuh, Fabry /Aufwandschätzung/ 191f.

Metrik wurde daher seit ihrer Veröffentlichung Ende der 1970er Jahre in einer Vielzahl von Varianten weiterentwickelt, die unterschiedliche Aspekte einer zu entwickelnden Software berücksichtigen, bestimmten angewandten Entwicklungsparadigmen Rechnung tragen sollen und spezifische Abbildungsregeln definieren.⁴⁰⁶ Den verschiedenen Varianten liegen somit unterschiedliche Verständnisse der maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand zugrunde.

Darüber hinaus zeigen Kitchenham, Pfleeger und Fenton auf, dass bei der Berechnung der Function Points unzulässige Operationen mit Größen unterschiedlicher Skalenniveaus durchgeführt werden. Der Ausdruck des Umfangs einer Software in Form von Function Points ist somit aus wissenschaftlicher Sicht umstritten.⁴⁰⁷

3.1.2 Weitere Einflussfaktoren auf den Entwicklungsaufwand

Es kann als allgemein akzeptiert angesehen werden, dass der Aufwand, der aus der Entwicklung einer Software eines bestimmten Umfangs resultiert, wesentlich dadurch beeinflusst wird, wie weitere Einflussfaktoren ausgeprägt sind. Da sich Entwicklungsvorhaben als komplexe, dynamische sozio-technische Systeme darstellen, scheint der Vielfalt von Einflussfaktoren kaum Grenzen gesetzt. So werden in der Fachliteratur Einflussfaktoren beschrieben, die sich auf die konzeptionellen Merkmalen der entwickelten Software, die verwendeten Technologien, die Projektmitarbeitern oder den herrschenden Rahmenbedingungen beziehen. Dabei werden sowohl Aspekte der Entwicklungs- und Unterstützungsaufgaben als auch der organisatorischen Gestaltungsmaßnahmen thematisiert. Manche Studien identifizieren 104 relevante Einflussfaktoren, andere münden gar in einem Katalog von 1.200 Einflussfaktoren.⁴⁰⁸

▪ Studie von Walston und Felix

Einen frühen Versuch, die maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand zu identifizieren, stellt die in den 1970er Jahren durch Walston und Felix durch-

⁴⁰⁶ Jones erwähnt diesbezüglich, dass mehr als 35 Varianten der Function Point Metrik veröffentlicht worden seien (vgl. Jones /Software Costs/ 272). Bundschuh und Fabry stellen einige dieser Varianten vergleichend dar (vgl. Bundschuh, Fabry /Aufwandschätzung/ 205-216). Zur Konzeption und Validierung einer an das Entwicklungsparadigma der Objektorientierung angepassten Function Point Metrik siehe beispielsweise Antoniol, Fiutem, Lokan /Object-Oriented Function Points/ sowie die dort referenzierten Quellen.

⁴⁰⁷ Vgl. Kitchenham, Pfleeger, Fenton /Measurement Validation/ 936.

⁴⁰⁸ Vgl. beispielsweise Conte, Dunsmore, Shen /Metrics and Models/ 238f.; Stevenson /Software Engineering Productivity/ 5f.; Noth, Kretzschmar /Aufwandschätzung/ 7.

geführte Studie bei IBM dar, deren Ergebnisse wiederholt in der Fachliteratur aufgegriffen werden.⁴⁰⁹

Unter Bezugnahme auf verschiedene Beiträge der Fachliteratur wurden mehrere Fragebögen konzipiert, anhand derer detaillierte Berichte von Projektmanagern zu 60 abgeschlossenen Softwareentwicklungsprojekten erfasst wurden. Diese Projekte erforderten einen Entwicklungsaufwand von 12 Personenmonaten bis zu 11.758 Personenmonaten. Die entwickelten Softwaresysteme umfassten 4.000 bis zu 467.000 Programmzeilen,⁴¹⁰ basierten auf unterschiedlichen Technologien und dienten unterschiedlichen Anwendungsgebieten.⁴¹¹

Auf der Grundlage dieses sehr heterogenen Datenbestands wurde zunächst ein nahezu linearer Zusammenhang zwischen der Anzahl der Programmzeilen der entwickelten Softwaresysteme und dem Entwicklungsaufwand festgestellt. Die untersuchten Datenpunkte wiesen jedoch eine gewisse Streuung auf. In einem nächsten Schritt sollten daher diejenigen Faktoren identifiziert werden, die in signifikantem Zusammenhang zu der Produktivität der Softwareentwicklung standen. Es wurde angenommen, dass die durch sie verursachten Unterschiede der Produktivität in den einzelnen Projekten für die Streuung der Datenpunkte verantwortlich seien.⁴¹² Zu diesem Zweck wurden 68 der in den Datenbeständen erfassten Attribute der Projekte für weitere Analysen ausgewählt. Ihre jeweilige Ausprägung war durch die Projektleiter größtenteils anhand von dreistufigen Ratingskalen beurteilt worden. Teilweise war lediglich angegeben worden, ob der entsprechende Aspekt auf das betrachtete Projekt zutraf oder nicht. Im Zuge der Analyse wurden die Projekte für jedes der Attribute in Gruppen mit einheitlichen Attributsausprägungen eingeteilt. Für jede dieser Gruppen wurde die Anzahl von Programmzeilen berechnet, die durchschnittlich in einem bestimmten Zeitraum entwickelt wurde. Aufgrund des Vergleichs dieser durchschnittlichen Produktivitäten wurde auf die Zusammenhänge zwischen dem Attribut und der Produktivität bzw. dem Entwicklungsaufwand geschlossen. Auf diese Weise wurden 29 signifikante Einflussfaktoren identi-

⁴⁰⁹ Vgl. Walston, Felix /Programming Measurement and Estimation/ 55-72; Conte, Dunsmore, Shen /Metrics and Models/ 243-248.

⁴¹⁰ Bei der Zählung der Programmzeilen wurden Kommentare einbezogen. Wiederverwendeter Programmcode wurde dagegen ausgeschlossen (vgl. Walston, Felix /Programming Measurement and Estimation/ 58).

⁴¹¹ Vgl. Walston, Felix /Programming Measurement and Estimation/ 55-57; Conte, Dunsmore, Shen /Metrics and Models/ 243.

⁴¹² Vgl. Walston, Felix /Programming Measurement and Estimation/ 62. Die Autoren verstehen die Produktivität der Mitarbeiter demnach als die Anzahl der Programmzeilen, die in einer bestimmten Zeit entwickelt werden.

fiziert. Diese Einflussfaktoren, die unterschiedenen Ausprägungen und die jeweilige durchschnittliche Produktivität der Projekte sind in Tabelle 3-2 dargestellt.⁴¹³

Einflussfaktor	Durchschnittliche Produktivität (Programmzeilen/ Personenmonat)		
	(1)	(2)	(3)
Die Komplexität der Benutzerschnittstelle	< Normal 500	Normal 295	> Normal 124
Die Beteiligung der Benutzer an der Bestimmung der Anforderungen	Keine 491	Etwas 267	Viel 205
Durch den Kunden veranlasste Änderungen des Entwurfs der Software	Wenige 297	- -	Viele 196
Die Erfahrung des Kunden mit dem Anwendungsgebiet des zu entwickelnden Softwaresystems	Keine 318	Etwas 340	Viel 206
Die allgemeine Erfahrung und Qualifizierung der Mitarbeiter	Gering 132	Durchschnittlich 257	Hoch 410
Der prozentuale Anteil der für die Implementierung zuständigen Mitarbeiter, die auch an der Analyse und dem Entwurf beteiligt waren	< 25 % 153	25 - 50 % 242	> 50 % 391
Die vorhergehende Erfahrung mit der eingesetzten Hardware	Minimal 146	Durchschnittlich 270	Umfangreich 312
Die vorhergehende Erfahrung mit der Programmiersprache	Minimal 122	Durchschnittlich 225	Umfangreich 385
Die vorhergehende Erfahrung mit Anwendungssystemen von gleichem oder größerem Umfang und Komplexität	Minimal 146	Durchschnittlich 221	Extensive 410
Das Verhältnis von der durchschnittlichen Anzahl der Mitarbeiter zur Dauer des Projektes (Personen/Monate)	< 0,5 305	0,5 – 0,9 310	> 0,9 173
Die gleichzeitige Entwicklung der Hardware	Nein 297	- -	Ja 177
Zugang zum Entwicklungscomputer nach spezieller Rückfrage	0 % 226	1 – 25 % 274	> 25 % 357
Kein Zugang zum Entwicklungscomputer	0 – 10 % 303	11 – 85 % 251	> 85 % 170
Bestimmte Sicherheitsanforderungen an Computer, Programme und Daten	Nein 289	- -	Ja 156
Strukturierte Programmierung	0 – 33 % 169	34 – 66 % -	66 % 301
Inspektionen von Entwurf und Programmcode	0 – 33 % 220	34 – 66 % 300	> 66 % 339
Top-Down-Entwurf	0 – 33 % 196	34 – 66 % 237	> 66 % 321

⁴¹³ Vgl. Walston, Felix /Programming Measurement and Estimation/ 64f.

Einflussfaktor	Durchschnittliche Produktivität (Programmzeilen/ Personenmonat)		
	(1)	(2)	(3)
Anwendung der Chief-Programmierer-Technik	0 – 33 % 219	34 – 66 % -	> 66 % 408
Die allgemeine Komplexität des entwickelten Programmcodes	< Durchschnittlich 314	-	> Durchschnittlich 185
Die Komplexität der Datenverarbeitung	< Durchschnittlich 349	Durchschnittlich 345	> Durchschnittlich 168
Die Komplexität des Kontrollflusses	< Durchschnittlich 289	Durchschnittlich 299	> Durchschnittlich 209
Die im Allgemeinen zu berücksichtigenden Randbedingungen beim Entwurf des Softwaresystems	Minimal 293	Durchschnittlich 286	Schwerwiegend 166
Hauptspeicherbegrenzungen	Minimal 391	Durchschnittlich 277	Schwerwiegend 193
Antwortzeitbedingungen	Minimal 303	Durchschnittlich 317	Schwerwiegend 171
Realtime, interaktive Operationen unter strengen Zeitbedingungen	< 10 % 279	10 – 40 % 337	> 40 % 203
Anteil der abgelieferten Statements	0 – 90 % 159	91 – 99 % 327	100 % 265
Anteil nicht-mathematischer und nicht Ein-/Ausgabe Programmzeilen	0 – 33 % 188	34 – 66 % 311	67 – 100 % 267
Anzahl der Kommentarzeilen pro 1000 Programmzeilen	0 – 15 334	16 – 80 243	> 80 193
Anzahl der Seiten der ausgelieferten Dokumentation pro 1000 Zeilen von ausgeliefertem Programmcode	0 – 32 320	33 – 88 252	> 88 195

Tabelle 3-2: Einflussfaktoren auf die Produktivität gemäß Walston und Felix⁴¹⁴

Auch wenn diese Ergebnisse der Studie von Walston und Felix in der Fachliteratur verschiedentlich aufgegriffen werden, erscheint ihre Interpretation aus verschiedenen Gründen als äußerst schwierig, von denen im Weiteren einige angeführt werden.⁴¹⁵

Beispielsweise bleibt das genaue Verständnis der einzelnen Einflussfaktoren im Unklaren und auch die Entscheidung zu ihrer Aufnahme in den Fragebogen und die Analyse wird nicht erläutert. Die Autoren geben an, dass die Ausprägungen der allgemeinen Erfahrung und Qualifizierung der Projektmitarbeiter, die Ausprägungen der allgemeinen

⁴¹⁴ Vgl. Walston, Felix /Programming Measurement and Estimation/ 64f.

⁴¹⁵ Vgl. Conte, Dunsmore, Shen /Metrics and Models/ 246-248; Walston, Felix /Programming Measurement and Estimation/ 63.

Komplexität des Programmcodes und die Ausprägungen der im Allgemeinen zu berücksichtigenden Randbedingungen beim Entwurf des Softwaresystems durch die Kombination der Angaben zu mehreren Fragen des Fragebogens gebildet wurden. Um welche Fragen es sich dabei handelt und wie die Kombination der zugehörigen Angaben durchgeführt wurde, wird nicht erläutert.

Die erfassten Daten beziehen sich darüber hinaus auf sehr unterschiedliche Projekte. Angesichts der begrenzten Anzahl der erfassten Projekte erscheint die Anzahl der analysierten Variablen als sehr umfangreich.

Schließlich geschah die Analyse der einzelnen Faktoren isoliert voneinander. Es wurde somit nicht berücksichtigt, dass möglicherweise Zusammenhänge zwischen den Einflussfaktoren bestehen oder Wechselwirkungen im Hinblick auf den Entwicklungsaufwand auftreten können.

▪ **COCOMO II**

Eine relativ große Verbreitung hat das sog. Constructive Cost Model (COCOMO) des Zentrums für Software Engineering (Center for Software Engineering, CSE) der Universität von Südkalifornien (University of Southern California, USC) erfahren. Dieses parametrische Verfahren zur Aufwandschätzung wurde 1981 von Boehm in seiner ursprünglichen Form vorgestellt und seit Mitte der 1990er Jahre in einer zweiten Version weiterentwickelt (COCOMO II).⁴¹⁶

Auf der Grundlage der Fachliteratur, ergänzt um die Erkenntnisse aus Tätigkeitsanalysen von Entwicklungsvorhaben und Expertenbefragungen, wurden verschiedene potenzielle Einflussfaktoren auf den Entwicklungsaufwand identifiziert. Diese wurden daraufhin aufgrund empirischer Daten und weiterer Expertenbefragungen in ein mathematisches Modell zur Vorhersage des Entwicklungsaufwands überführt. Eine detaillierte Erläuterung dieser einzelnen Schritte ist jedoch nicht bekannt. Die intersubjektive Verständlichkeit des Untersuchungsprozesses und der Beurteilung der Untersuchungsergebnisse ist daher nicht gewährleistet.⁴¹⁷

Ausgehend vom Umfang der zu entwickelnden Software, verstanden als die Anzahl der Programmzeilen, werden verschiedene Einflussfaktoren in die Prognose des erforderli-

⁴¹⁶ Vgl. Boehm /Software Engineering Economics/; Boehm u. a. /COCOMO II/.

⁴¹⁷ Vgl. Boehm u. a. /COCOMO II/ 143.

chen Entwicklungsaufwands einbezogen. Einige von ihnen werden als sog. Skalierungsfaktoren (scaling drivers) angesehen. Ihre Ausprägung wird in Verbindung mit dem Umfang der Software zur Bestimmung des grundlegenden, sog. nominalen Entwicklungsaufwands herangezogen. Aufbauend auf diesem nominalen Entwicklungsaufwand werden weitere Einflussfaktoren in die Prognose einbezogen, die sich auf die Software, die Projektmitarbeiter und sonstige Aspekte des Projekts beziehen. Sie werden als „effort multiplifier“, im Weiteren übersetzt als Aufwandstreiber, bezeichnet.

Wie eingangs erläutert, kann eine Aufwandschätzung zu verschiedenen Zeitpunkten im Verlauf eines Entwicklungsvorhabens durchgeführt werden.⁴¹⁸ Diesbezüglich wurde ebenfalls darauf hingewiesen, dass mit dem fortschreitenden Entwicklungsvorhaben zunehmend genaue Informationen zu der Ausprägung der Einflussfaktoren vorliegen. Diesem Sachverhalt trägt das COCOMO II-Verfahren Rechnung, indem zwei unterschiedliche Modelle der Aufwandstreiber verwendet werden.

Das erste Modell soll die Aufwandschätzung zu einem Zeitpunkt ermöglichen, da lediglich ein grober Entwurf der zu entwickelnden Software durchgeführt und die letztendlich gültige Architektur noch nicht festgelegt wurde. Da zu diesem Zeitpunkt nur wenige Informationen zum Entwicklungsvorhaben vorliegen, werden nur relativ wenige Einflussfaktoren in dem Modell abgebildet (Early-Design Model). Das zweite Modell geht hingegen von einem Zeitpunkt der Aufwandschätzung aus, an dem die Architektur des zu entwickelnden Softwaresystems vollständig konzipiert wurde und mehr Informationen für die Schätzung zur Verfügung stehen. Dem wird mit einer differenzierteren Abbildung der Einflussfaktoren entsprochen (Post-Architecture Model).⁴¹⁹

Die Einflussfaktoren, die als Aufwandstreiber in diesem zweiten, differenzierteren Modell abgebildet sind, werden gemeinsam mit den zuvor angesprochenen Skalierungsfaktoren in Tabelle 3-3 dargestellt. Für die Beurteilung ihrer Ausprägungen werden im COCOMO II-Verfahren jeweils vier- bis sechsstufige Skalen angegeben.⁴²⁰

⁴¹⁸ Vgl. Kapitel 1.4.4.

⁴¹⁹ Vgl. CSE-USC /COCOMO II/ 2.

⁴²⁰ Vgl. CSE-USC /COCOMO II/ 5-9, 26-33. Die gewählten Übersetzungen für die englischsprachigen Bezeichnungen der Einflussfaktoren orientieren sich vorwiegend an Balzert /Software-Technik/ 91f. Auf die Darstellung der im Early-Design Model abgebildeten Einflussfaktoren wird an dieser Stelle verzichtet, da es sich dabei lediglich um Zusammenfassungen der im Post-Architecture Model abgebildeten Faktoren handelt oder diese teilweise unmittelbar übernommen wurden (vgl. Agarwal u. a. /Estimating/ 63). Aktuellere Überlegungen zur Erweiterung des Modells, wie beispielsweise um die Anforderungen an die Sicherheit des Anwendungssystems als weiteren Einflussfaktor auf den Entwicklungsaufwand, werden an dieser Stelle nicht aufgegriffen (vgl. Reifer, Boehm, Gangadharan /Cost of Security/ 181-183).

Einflussfaktor	Erläuterung
Skalierungsfaktoren	
Neuartigkeit	<p>Dieser Skalierungsfaktor repräsentiert mehrere Einflussfaktoren auf den Entwicklungsaufwand:</p> <ul style="list-style-type: none"> – das organisatorische Verständnis der mit dem Anwendungssystem verfolgten Ziele, – die Erfahrung im Umgang mit ähnlichen Anwendungssystemen, – die parallele Entwicklung von Hardwarekomponenten oder Arbeitsabläufen, die mit dem Anwendungssystem in Zusammenhang stehen, sowie – die Notwendigkeit innovative Architekturen oder Algorithmen zu entwickeln.
Entwicklungsflexibilität	<p>Dieser Skalierungsfaktor repräsentiert mehrere Einflussfaktoren:</p> <ul style="list-style-type: none"> – die Notwendigkeit, vorher festgelegte Anforderungen einzuhalten, – die Notwendigkeit, bestimmte Spezifikationen der externen Schnittstellen einzuhalten, sowie – der Anreiz für eine frühzeitige Fertigstellung der Entwicklung.
Klärung der Architektur und damit verbundener Risiken	<p>Dieser Skalierungsfaktor repräsentiert mehrere Einflussfaktoren:</p> <ul style="list-style-type: none"> – das Ausmaß, in dem der Risikomanagementplan alle kritischen Architekturrisiken identifiziert und Meilensteine für die Auflösung dieser Risiken im Zuge eines Architekturreviews festlegt, – die Kompatibilität des Zeitplans, des Budgets und der internen Meilensteine mit dem Risikomanagementplan infolge eines Architekturreviews, – der Anteil der Entwicklungszeit, der bei durchschnittlichen Entwicklungszielen der Festlegung der Architektur des Anwendungssystems gewidmet ist, – die Verfügbarkeit der erforderlichen hoch qualifizierten Mitarbeiter für die Gestaltung der Architektur des Anwendungssystems, – die Unterstützung der Auflösung von Architekturrisiken, der Entwicklung und der Verifikation von Architekturspezifikationen durch Entwicklungswerkzeuge, – das Ausmaß der Unsicherheit bezüglich wesentlicher Einflussfaktoren auf die Gestaltung der Architektur des Anwendungssystems (z. B. Entwicklungsziele, Hardware, Technologien, verfügbare Standardsoftwarekomponenten) sowie – die Anzahl kritischer Architekturrisiken.
Zusammenhalt der Projektbeteiligten	<p>Dieser Skalierungsfaktor repräsentiert mehrere Einflussfaktoren:</p> <ul style="list-style-type: none"> – die Übereinstimmung der Ziele und Kulturen der Projektbeteiligten, – die Möglichkeit und Bereitschaft der Projektbeteiligten, die Ziele anderer Projektbeteiligter zu unterstützen, – die Erfahrung der Projektbeteiligten, als Team zu arbeiten, sowie – die Investition in gemeinsame Visionen und Bekenntnisse der Projektbeteiligten.
Reifegrad des Entwicklungsprozesses	<p>Der Reifegrad des Entwicklungsprozesses wird in Anlehnung an das Capability Maturity Model (CMM) und die darin aufgeführten entscheidenden Prozessbereiche (engl. key process areas) bestimmt.</p>
Anwendungssystembezogene Faktoren	
Zuverlässigkeit	<p>Dieser Einflussfaktor bezeichnet die erforderliche Zuverlässigkeit des Anwendungssystems bzw. die Schwere der Folgen eines Fehlverhaltens.</p>
Größe der zu verwaltenden Datenbestände	<p>Dieser Einflussfaktor bezeichnet die Größe, der im Anwendungssystem zu verwaltenden Datenbestände.</p>

Einflussfaktor	Erläuterung
Komplexität	Dieser Einflussfaktor repräsentiert die Komplexität des Anwendungssystems. Diese wird anhand verschiedener Operationen des Systems bestimmt: <ul style="list-style-type: none"> – die Operationen zur Steuerung des Kontrollflusses, – die Operationen zur Datenverarbeitung, – die Operationen zur Steuerung von Hardwarekomponenten, – die Operationen zur Datenverwaltung, sowie – die Operationen zur Verwaltung der Benutzerschnittstelle.
Wiederverwendbarkeit	Dieser Einflussfaktor bezeichnet die Anforderungen an die Wiederverwendbarkeit der Komponenten des Anwendungssystems.
Dokumentation	Dieser Einflussfaktor bezeichnet die Anforderungen an die Dokumentation des Anwendungssystems, insbesondere bezüglich der Anpassung der Dokumentation an spezifische Bedürfnisse einzelner Lebenszyklusabschnitte des Anwendungssystems.
Plattformbezogene Faktoren	
Randbedingungen bezogen auf die Ausführungszeit	Dieser Einflussfaktor bezeichnet die zu berücksichtigenden Randbedingungen bezüglich der verfügbaren Ausführungszeit infolge des Auslastungsgrades der verfügbaren Ressourcen.
Randbedingungen bezogen auf den Arbeitsspeicher	Dieser Einflussfaktor bezeichnet die zu berücksichtigenden Randbedingungen bezüglich des verfügbaren Arbeitsspeichers infolge des Auslastungsgrades der verfügbaren Ressourcen.
Stabilität der Plattform	Dieser Einflussfaktor bezeichnet die Änderungshäufigkeit der Hardware- und Softwarekomponenten der Plattform des Anwendungssystems.
Personalbezogene Faktoren	
Fähigkeiten der mit Analyse und Entwurf betrauten Mitarbeiter	Dieser Einflussfaktor repräsentiert die Fähigkeiten der mit Analyse und Entwurf betrauten Mitarbeiter. Diese werden im Hinblick auf verschiedene Aspekte bestimmt: <ul style="list-style-type: none"> – die Fertigkeiten der Mitarbeiter hinsichtlich Analyse und Entwurf, – die Effizienz und Genauigkeit ihrer Arbeit, – ihre Fähigkeiten hinsichtlich der Kommunikation und Kooperation.
Fähigkeiten der mit der Implementierung betrauten Mitarbeiter	Dieser Einflussfaktor repräsentiert die Fähigkeiten der mit der Implementierung betrauten Mitarbeiter. Diese werden im Hinblick auf verschiedene Aspekte bestimmt: <ul style="list-style-type: none"> – die Fertigkeiten der Mitarbeiter hinsichtlich der Implementierung, – die Effizienz und Genauigkeit ihrer Arbeit, – ihre Fähigkeiten hinsichtlich der Kommunikation und Kooperation.
Erfahrung mit der Anwendung	Dieser Einflussfaktor bezeichnet das Ausmaß der Erfahrung der Projektmitarbeiter mit der entsprechenden Art von Anwendungssystemen.
Erfahrung mit der Plattform	Dieser Einflussfaktor bezeichnet das Ausmaß der Erfahrung der Projektmitarbeiter mit leistungsfähigen Plattformen von Anwendungssystemen.
Erfahrung mit Programmiersprachen und Entwicklungswerkzeugen	Dieser Einflussfaktor bezeichnet das Ausmaß der Erfahrung der Projektmitarbeiter im Umgang mit den eingesetzten Programmiersprachen und Entwicklungswerkzeugen.
Kontinuität der Mitarbeiter	Dieser Einflussfaktor bezeichnet die Dauer und Durchgängigkeit der Mitarbeit der Personen in dem Entwicklungsvorhaben.
Projektbezogene Faktoren	
Einsatz von Entwicklungswerkzeugen	Dieser Einflussfaktor bezeichnet die Unterstützung der Entwicklung durch Entwicklungswerkzeuge.

Einflussfaktor	Erläuterung
Entwicklung an mehreren Standorten	Dieser Aufwandstreiber repräsentiert mehrere Einflussfaktoren: <ul style="list-style-type: none">– die Verteilung der Entwicklung auf verschiedene Standorte,– die verfügbaren Kommunikationsmittel.
Zulässige Entwicklungsdauer	Dieser Einflussfaktor bezeichnet die zu berücksichtigenden Randbedingungen bezüglich der Entwicklungsdauer.

Tabelle 3-3: Einflussfaktoren auf den Entwicklungsaufwand im COCOMO II-Verfahren⁴²¹

Dem COCOMO II-Verfahren liegen verschiedene Annahmen zugrunde. Beispielsweise wird vorausgesetzt, dass sich die definierten Anforderungen im Verlauf des Entwicklungsvorhabens nicht mehr substantiell ändern. Es wird darauf verwiesen, dass tief greifende Modifikationen oder Erweiterungen der Anforderungen eine erneute Schätzung des Entwicklungsaufwands aufgrund der veränderten Sachverhalte erforderlich machen. Die Stabilität der Anforderungen wird somit nicht als Einflussfaktor auf den Entwicklungsaufwand in dem Verfahren berücksichtigt.⁴²²

Des Weiteren basiert das COCOMO II-Verfahren auf der Annahme, dass ein effektives Projektmanagement gewährleistet ist und somit beispielsweise der Anteil der nicht produktiv eingesetzten Arbeitszeit aller Projektbeteiligten gering gehalten wird. Managementbezogene Aspekte werden somit nicht als Einflussfaktoren auf den Entwicklungsaufwand berücksichtigt.⁴²³ Gerade diese managementbezogenen Aspekte werden jedoch vielfach als zentrale Determinanten des Entwicklungsaufwands angesehen.⁴²⁴ So hebt Gilb hervor, dass sich die meisten Möglichkeiten zur Erhöhung der Produktivität der Softwareentwicklung auf das Management und nicht auf die Technik bezögen.⁴²⁵

3.2 Erkenntnisse zum Einsatz von Standardsoftwarekomponenten

In zunehmendem Maße werden in der Entwicklung von Softwaresystemen bereits bestehende Softwarekomponenten eingesetzt. Häufig handelt es sich dabei nicht um die Wiederverwendung zuvor entwickelter Software, sondern um kommerziell erworbene Standardsoftwarekomponenten, die in das Softwaresystem eingebaut werden. Diese werden im Englischen oftmals als „commercial-off-the-shelf (COTS) software compo-

⁴²¹ Vgl. CSE-USC /COCOMO II/ 5-9, 26-33. Die gewählten Übersetzungen für die englischsprachigen Bezeichnungen der Einflussfaktoren orientieren sich vorwiegend an Balzert /Software-Technik/ 91f.

⁴²² Vgl. Boehm /Software Engineering Economics/ 59.

⁴²³ Vgl. Boehm /Software Engineering Economics/ 59.

⁴²⁴ Vgl. Stevenson /Software Engineering Productivity/ 541f.

⁴²⁵ Vgl. Gilb /Software Engineering Management/ 256f.

nents“ bezeichnet.⁴²⁶ Im Rahmen der Erläuterung der Integrationsprodukte wurde bereits darauf hingewiesen, dass sich der Einsatz der kommerziellen Standardsoftwarekomponenten durch besondere Merkmale auszeichnet.⁴²⁷ Insbesondere ist der Quellcode der Standardsoftwarekomponenten oftmals nicht zugänglich und man hat im Allgemeinen keine Kontrolle über ihre funktions- und qualitätsbezogenen Merkmale sowie deren Weiterentwicklung durch die Hersteller.⁴²⁸

Am Zentrum für Software Engineering der Universität von Südkalifornien wird seit 1996 ein parametrisches Verfahren zum Schätzen des Aufwands für die Entwicklung eines Softwaresystems unter Verwendung kommerziell erworbener Standardsoftwarekomponenten entwickelt. Dieses Constructive COTS Integration Cost Model (COCOTS) soll zur Ergänzung des bereits erläuterten COCOMO II-Verfahrens dienen und bezieht sich daher lediglich auf die Aspekte der Softwareentwicklung, die unmittelbar mit dem Einsatz der Standardsoftwarekomponenten zusammenhängen.⁴²⁹

Auf der Grundlage der Fachliteratur, der Auswertung von Untersuchungsergebnissen des Software Engineering Institute der Carnegie Mellon University zu Projektrisiken und bereits existierender Verfahren der Aufwandschätzung (z. B. dem COTS Integration Cost Model von Loral Federal Systems) wurden verschiedene potenzielle Einflussfaktoren auf den Entwicklungsaufwand identifiziert. Diese wurden um Erkenntnisse aus Tätigkeitsanalysen von Entwicklungsvorhaben und Expertenbefragungen ergänzt. Gegenwärtig wird aufgrund empirischer Daten ein mathematisches Modell zur Vorhersage des Entwicklungsaufwands aufgestellt. Wie beim COCOMO II-Verfahren kann jedoch wiederum bemängelt werden, dass die einzelnen Schritte nicht angemessen erläutert werden, um die intersubjektive Verständlichkeit des Untersuchungsprozesses und der Beurteilung der Untersuchungsergebnisse zu gewährleisten.⁴³⁰

Das COCOTS-Verfahren beruht auf dem Verständnis, dass der Einsatz von Standardsoftwarekomponenten mit vier Teilaufgaben verbunden ist: die Beurteilung und Auswahl der Standardsoftwarekomponenten, die Konfiguration und Anpassung der Standardsoftwarekomponenten, die Entwicklung des für die Einbindung der Standardsoft-

⁴²⁶ Vgl. Abts, Boehm /COCOTS/ 3.

⁴²⁷ Vgl. Kapitel 2.2.4.4.

⁴²⁸ Vgl. Abts, Boehm, Clark /COCOTS/ 1.

⁴²⁹ Vgl. Abts, Boehm, Clark /COCOTS/ 1f.; Boehm u. a. /COCOMO II/ 237-253.

⁴³⁰ Vgl. Abts, Boehm, Clark /COCOTS/ 2f.; Abts, Boehm /COCOTS/ 19-22.

warekomponenten erforderlichen Programmcodes (glue code) und die Anpassung anderer Komponenten des Softwaresystems.⁴³¹ Für den mit den einzelnen Teilaufgaben verbundenen Aufwand werden unterschiedliche Einflussfaktoren als maßgeblich angesehen.⁴³²

Für den Aufwand für die Beurteilung und Auswahl der einzusetzenden Standardkomponenten wird zunächst die Anzahl der betrachteten Arten von Standardkomponenten als wichtiger Einflussfaktor angesehen. Die Einteilung der Standardsoftwarekomponenten in Arten erfolgt dabei anhand ihrer grundlegenden funktionsbezogenen Merkmale. Darauf aufbauend werden die Anzahl der alternativen Standardsoftwarekomponenten innerhalb dieser Gruppen sowie der für die Beurteilung und Auswahl einer Softwarekomponente einer bestimmten Art durchschnittlich erforderliche Aufwand betrachtet. Dies bedeutet, dass der für diese Teilaufgabe erforderliche Aufwand im COCOTS-Verfahren lediglich als die Summe des Aufwands je Standardsoftwarekomponente bestimmt wird. Somit können lediglich die Anzahl und die Art der Standardsoftwarekomponenten als Einflussfaktoren im Sinne dieser Arbeit interpretiert werden.⁴³³

Ein ähnliches Verständnis liegt der Prognose des Aufwands für die Konfiguration und Anpassung der Standardsoftwarekomponenten zugrunde. Wiederum werden die Anzahl und die Art der Standardsoftwarekomponenten erfasst. Darauf aufbauend wird der mit der Teilaufgabe verbundene Aufwand wiederum als die Summe des Aufwands je Standardsoftwarekomponente verstanden. Dabei wird jedoch auch die Unterstützung der Konfiguration und Anpassung durch Entwicklungswerkzeuge betrachtet. Die Anzahl und die Art der Standardsoftwarekomponenten sowie das Ausmaß der Unterstützung durch Entwicklungswerkzeuge können somit als Einflussfaktoren auf den Aufwand interpretiert werden.⁴³⁴

Differenzierter stellt sich das COCOTS-Verfahren im Hinblick auf den für die Einbindung der Standardsoftwarekomponenten zu entwickelnden Programmcode dar. Dieser wird abgegrenzt als individuell entwickelter Programmcode, der die Interaktion einer Standardsoftwarekomponente mit anderen Komponenten des Softwaresystems unterstützt, der eine Standardsoftwarekomponente im Softwaresystem verankert oder der die

⁴³¹ Vgl. Abts, Boehm, Clark /COCOTS/ 3.

⁴³² Vgl. Abts, Boehm, Clark /COCOTS/ 9.

⁴³³ Vgl. Abts, Boehm, Clark /COCOTS/ 5f.

⁴³⁴ Vgl. Abts, Boehm, Clark /COCOTS/ 6f.

Funktionalität einer Standardsoftwarekomponente ergänzt und damit mit dieser interagieren muss bzw. von ihr abhängt.⁴³⁵ Der Umfang des diesbezüglich zu entwickelnden Programmcodes wird durch die Anzahl der Programmzeilen erfasst. Er wird darüber hinaus um den Anteil erforderlicher Überarbeitungen infolge von Veränderungen der Anforderungen oder von Weiterentwicklungen der eingesetzten Standardsoftwarekomponenten erweitert.⁴³⁶

Aufbauend auf diesem Verständnis des Umfangs des für die Einbindung der Standardsoftwarekomponenten entwickelten Programmcodes werden 13 weitere Einflussfaktoren angeführt, die den Aufwand für diese Teilaufgabe beeinflussen sollen. Sie sind in Tabelle 3-4 dargestellt. Im Hinblick auf diese Einflussfaktoren erscheint es als kritisch, dass sich die gegenwärtig im COCOTS-Verfahren abgebildeten Einflussfaktoren vielfach von den in früheren Veröffentlichungen erläuterten Faktoren unterscheiden. Die Grundlagen für diese Veränderungen ebenso wie Erläuterungen zum Verständnis der neu aufgenommenen oder verändert begriffenen Einflussfaktoren und ihre Messung sind nicht bekannt. Die Interpretation der im COCOTS-Verfahren dargestellten Erkenntnisse zum Aufwand für die Entwicklung des Programmcodes zur Einbindung der Standardsoftwarekomponenten ist angesichts dieser beschränkten Informationen nur eingeschränkt möglich.⁴³⁷

⁴³⁵ Vgl. Abts, Boehm, Clark /COCOTS/ 7.

⁴³⁶ Vgl. Abts, Boehm, Clark /COCOTS/ 7f.

⁴³⁷ Vgl. Abts, Boehm, Clark /COCOTS/ 8; Abts, Boehm /COCOTS/ 34-38.

Einflussfaktor	Erläuterung
Personenbezogene Faktoren	
Erfahrung mit der Standardsoftwarekomponente	Dieser Einflussfaktor bezeichnet die Erfahrung der zuständigen Projektmitarbeiter mit der Einbindung, dem Betrieb und der Wartung der betreffenden Standardsoftwarekomponente.
Generelle Fähigkeiten der zuständigen Projektmitarbeiter im Hinblick auf die Softwareentwicklung	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Erfahrung mit dem Vorgehen zur Integration von Standardsoftwarekomponenten	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Kontinuität der zuständigen Projektmitarbeiter.	Eine Definition dieses Einflussfaktors ist nicht bekannt. In Anlehnung an das COCOMO II-Verfahren könnte dieser Einflussfaktor die Dauer und Durchgängigkeit der Mitarbeit der Personen an der Integration der Standardsoftwarekomponenten bezeichnen.
Standardsoftwarekomponentenbezogene Faktoren	
Reifegrad der Standardsoftwarekomponente	Dieser Einflussfaktor bezeichnet, wie lange die Standardsoftwarekomponente bereits am Markt angeboten wurde, wie häufig sie bereits verkauft wurde und ob sie bereits einen guten Ruf im Hinblick auf ihren Nutzen und ihre Zuverlässigkeit erworben hat.
Bereitschaft des Herstellers, die Standardsoftwarekomponente anzupassen	Dieser Einflussfaktor bezeichnet die Bereitschaft des Herstellers der Standardsoftwarekomponente diese an spezifische Bedürfnisse anzupassen.
Komplexität der Schnittstelle der Standardsoftwarekomponente	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Unterstützung durch den Hersteller	Dieser Einflussfaktor bezeichnet die technische Unterstützung, die der Hersteller der Standardsoftwarekomponente seinen Kunden bietet, beispielsweise durch eigene Mitarbeiter oder durch Drittunternehmen.
Durch den Hersteller angebotene Schulung und Dokumentation	Dieser Einflussfaktor bezieht sich auf zwei Aspekte. Zum einen betrifft er den Umfang der durch den Hersteller angebotenen Schulungen im Umgang mit einer Standardsoftwarekomponente, die er beispielsweise durch eigene Mitarbeiter oder durch Drittunternehmen durchführen kann. Zum anderen betrifft er Umfang und Qualität der mit der Standardsoftwarekomponente gelieferten Dokumentation.
Anwendungssystembezogene Faktoren	
Randbedingungen bezogen auf die Zuverlässigkeit des Anwendungssystems	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Komplexität der Schnittstelle des Anwendungssystems	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Randbedingungen bezogen auf die Performanz der Standardsoftwarekomponente	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Portabilität des Anwendungssystems	Eine Definition dieses Einflussfaktors ist nicht bekannt.
Skalierungsfaktoren	
Entwicklung der Architektur des Anwendungssystems	Eine Definition dieses Einflussfaktors ist nicht bekannt.

Tabelle 3-4: Einflussfaktoren auf den Aufwand für die Entwicklung des Programmcodes für die Integration von Standardsoftwarekomponenten gemäß dem COCOTS-Verfahren⁴³⁸

⁴³⁸ Vgl. Abts, Boehm, Clark /COCOTS/ 8; Abts, Boehm /COCOTS/ 34-38.

Für den Aufwand, der aus Anpassungen anderer Komponenten des Softwaresystems resultiert, bietet das COCOTS-Verfahren keine weiterführenden Erkenntnisse zu maßgeblichen Einflussfaktoren.

3.3 Erkenntnisse zur Entwicklung eines Systems von Softwaresystemen

Eine zunehmende Bedeutung wird solchen Softwaresystemen beigemessen, die als „Systems of Systems (SoS)“, d. h. Systeme von Softwaresystemen charakterisiert werden können. Sie bestehen aus mehreren Subsystemen, die unabhängig voneinander betrieben werden und neben ihrer Funktion im Gesamtsystem zum Erreichen eigenständiger Zwecke eingesetzt werden. Oftmals werden die einzelnen Subsysteme durch unterschiedliche Organisationen betrieben und sind damit geographisch stark verteilt. Ein SoS stellt mehr als nur die Summe seiner Teilsysteme dar. Es weist ein Verhalten auf, das nicht in den einzelnen Subsystemen zu finden ist. Es wird dadurch charakterisiert, dass kontinuierlich Teile hinzugefügt, verändert oder entfernt und damit Aufbau, Funktionalität und Zweck des SoS evolutionär weiterentwickelt werden.⁴³⁹

Wiederum wird am Zentrum für Software Engineering der Universität von Südkalifornien an einem parametrischen Verfahren zur Aufwandschätzung entwickelt, das in Ergänzung des erläuterten COCOMO II-Verfahrens eingesetzt werden soll. Dieses Constructive System of Systems Integration Model (COSOSIMO) bezieht sich damit lediglich auf die Aspekte der Softwareentwicklung, die unmittelbar mit der Integration der Subsysteme zusammenhängen. Seine Entwicklung basiert wiederum auf der bereits beschriebenen Vorgehensweise, ist jedoch weniger weit fortgeschritten als im Falle des COCOTS-Verfahrens. Auf der Grundlage der Fachliteratur, Tätigkeitsanalysen von Entwicklungsvorhaben und Expertenbefragungen wurden verschiedene potenzielle Einflussfaktoren auf den Entwicklungsaufwand identifiziert, deren Bedeutung gegenwärtig durch weitere Expertenbefragungen und erste quantitative Studien überprüft wird.⁴⁴⁰ Die gegenwärtig verfügbaren Informationen zum Entwicklungsstand des COSOSIMO-Verfahrens geben damit lediglich das vorläufige Verständnis der Einflussfaktoren auf den Entwicklungsaufwand von SoS wieder. Dabei wurde das wissenschaftliche Vorge-

⁴³⁹ Vgl. Sage, Cuppan /Systems of Systems/ 326f.

⁴⁴⁰ Vgl. Lane /System-of-Systems/ 1-3.

hen, auf denen das Modell beruht, bislang nicht in angemessener Weise transparent gemacht.⁴⁴¹ Eine theoretische oder empirische Validierung des aufgestellten Modells ist darüber hinaus nicht bekannt. Aufgrund des thematischen Bezugs zur Anwendungsintegration soll dennoch dieses vorläufige Verständnis der Einflussfaktoren dargestellt werden.

Ebenso wie das bereits behandelte COCOMO II-Verfahren liegen dem COSOSIMO-Verfahren zwei unterschiedliche Modelle der Einflussfaktoren zugrunde, eines zur Durchführung einer Aufwandschätzung zum Zeitpunkt des groben Entwurfs (Early-Design Model) und eines zur Durchführung einer Aufwandschätzung nach der vollständigen Konzeption der Architektur des zu entwickelnden SoS (Post-Architecture Model).⁴⁴²

Beiden Modellen liegt wiederum das Verständnis zugrunde, dass der Umfang des zu entwickelnden Softwaresystems den grundlegenden Einflussfaktor auf den Entwicklungsaufwand darstellt, dessen Wirkungsweise durch verschiedene weitere Einflussfaktoren beeinflusst wird. Sie unterscheiden sich jedoch darin, welche Faktoren den Umfang eines SoS determinieren und welche Einflussfaktoren darüber hinaus ausschlaggebend für den daraus resultierenden Entwicklungsaufwand sind. Die beiden Modelle beziehen sich diesbezüglich nicht nur auf unterschiedliche Zeitpunkte für die Durchführung einer Aufwandschätzung und damit zusammenhängend unterschiedliche verfügbare Informationen. Sie stellen auch unterschiedliche Momentaufnahmen im Erkenntnisprozess der maßgeblichen Einflussfaktoren auf den Aufwand für die Entwicklung eines SoS dar, wie im Weiteren verdeutlicht wird.

Aufgrund der ursprünglichen Annahmen zu den maßgeblichen Einflussfaktoren und ihren Wirkungsbeziehungen wurde zunächst das Post-Architecture Model aufgestellt. In diesem wird der Umfang des zu entwickelnden SoS anhand mehrerer Faktoren bestimmt, welche sich auf die Subsysteme des SoS oder deren Beziehungen zueinander beziehen. Dies ist zunächst der Umfang der schnittstellenbezogenen Software, der als Resultat der Anzahl der Programmzeilen, der Komplexität des Programmcodes, der Änderungsdynamik des Programmcodes und dem Anteil des wiederverwendeten Pro-

⁴⁴¹ So erwähnt Lane lediglich eine Literaturanalyse sowie Arbeitsgruppen und Workshops mit Experten ohne das jeweilige Vorgehen oder beispielsweise den Erfahrungshintergrund der beteiligten Experten zu erläutern (vgl. Lane /System-of-Systems/ 3-5).

⁴⁴² Vgl. Lane /System-of-Systems/ 5.

grammcodes der schnittstellenbezogenen Software verstanden wird. Des Weiteren wird der Umfang anhand der Anzahl der verwendeten Schnittstellen bestimmt. Dabei werden sowohl Schnittstellen zwischen den Subsystemen des SoS (externe Schnittstellen der Subsysteme) als auch Schnittstellen zwischen den verschiedenen Komponenten innerhalb eines Subsystems (interne Schnittstellen der Subsysteme). Als letzter Faktor zur Charakterisierung des Umfangs des zu entwickelnden SoS wird die Anzahl der Anwendungsszenarien angesehen, die durch das SoS unterstützt werden müssen (beispielsweise definiert als sog. Use Cases).⁴⁴³

Aufbauend auf diesem Verständnis des Umfangs eines SoS werden im Post-Architecture Model verschiedene weitere Einflussfaktoren in die Prognose des Entwicklungsaufwands einbezogen. Diese werden in Tabelle 3-5 dargestellt.⁴⁴⁴

Einflussfaktor	Erläuterung
Einfachheit der Integration	Dieser Faktor wird als Kombination verschiedener Sachverhalte verstanden. Im Einzelnen repräsentiert er das Ausmaß der Kopplung zwischen den Subsystemen des SoS, die Wichtigkeit der Verarbeitungsprozesse, die Reichweite der angestrebten Leistungsmerkmale und die Neuartigkeit des SoS.
Klärung von Integrationsrisiken	Dieser Faktor wird als Kombination verschiedener Sachverhalte verstanden. Im Einzelnen repräsentiert er die Anzahl bedeutender Integrationsrisiken, den Reifegrad des Risikomanagements ebenso wie die Kompatibilität von Plänen und Budgets, die Verfügbarkeit von Experten, die Unterstützung durch Entwicklungswerkzeuge sowie das Ausmaß an Unsicherheit im Hinblick auf die Integrationsrisiken.
Stabilität der Integration	Dieser Faktor bezeichnet das Ausmaß der Veränderungen der Subsysteme im Verlauf der Entwicklung des SoS.
Bereitschaft der Subsysteme für die Integration	Dieser Faktor wird zum einen durch das Ausmaß der Maßnahmen zur Verifikation und Validierung der Subsysteme bestimmt, die vor der Integration der Subsysteme durchgeführt werden. Zum anderen bezieht er sich auf das Ausmaß der Integrationsmaßnahmen im Hinblick auf Subsysteme, bevor diese in das SoS eingebunden werden.
Fähigkeiten des Integrations-teams	Dieser Faktor bezeichnet das Ausmaß der Kooperation und des Zusammenhalts unter den Mitarbeitern, die Fähigkeiten der Mitarbeiter, die Kontinuität der Mitarbeiter sowie die Erfahrung der Mitarbeiter mit der Anwendungen, den Programmiersprachen, den Integrationsprodukten und Integrationsplattformen.
Reifegrad des Integrationsprozesses	Dieser Faktor bezeichnet den Reifegrad und die Vollständigkeit der Integrationsprozesse, der Pläne sowie der Entwicklungsumgebung.

Tabelle 3-5: Einflussfaktoren auf den Aufwand für die Entwicklung von SoS gemäß des Post-Architecture Model des COSOSIMO-Verfahrens⁴⁴⁵

⁴⁴³ Vgl. Lane /System-of-System Integration/ 1. Dass sich die Anzahl der Schnittstellen sowohl auf die externen Schnittstellen als auch die internen Schnittstellen der Subsysteme bezieht, wird aus dem im Internet bereitgestellten Fragebogen zur Beurteilung des Post-Architecture Model ersichtlich. Dort werden darüber hinaus der Aufwand für die Entwicklung der einzelnen Subsysteme sowie die Anzahl der Subsysteme eines SoS als weitere Faktoren zur Charakterisierung der Größe des SoS aufgeführt (vgl. CSE-USC /Initial Survey/ 3).

⁴⁴⁴ Vgl. Lane /System-of-Systems/ 4.

In Workshops sowie einer schriftlichen Befragung sollte im nächsten Schritt gemeinsam mit erfahrenen Praktikern und Wissenschaftlern geklärt werden, ob die im Post-Architecture Model abgebildeten Einflussfaktoren tatsächlich als maßgeblich für den Entwicklungsaufwand angenommen werden können und inwiefern sie in dem COSOSIMO-Verfahren berücksichtigt werden sollten. Auch wenn ein großer Anteil der Befragten die Einflussfaktoren als relevant ansahen, wurde kritisiert, dass ihre Ausprägungen zu frühen Zeitpunkten im Entwicklungsvorhaben nicht absehbar seien und darüber hinaus weitere Einflussfaktoren in das Modell aufgenommen werden sollten. Aufgrund dieser Rückmeldungen wurde daher das Early-Design Model als alternatives Modell maßgeblicher Einflussfaktoren entwickelt, ohne das Post-Architecture Model zu verändern.⁴⁴⁶

Im Early-Design Model wird der Umfang des zu entwickelnden SoS zum einen durch die Anzahl der unterschiedlichen Schnittstellenprotokolle und zum anderen durch die Anzahl der Organisationen, die Komponenten des SoS beisteuern, erfasst. Beide dieser Faktoren werden darüber hinaus hinsichtlich ihrer Komplexität beurteilt. Die Komplexität der unterschiedlichen Schnittstellenprotokolle wird beispielsweise durch die Anzahl der Protokollschichten, die erforderliche Sicherheit und dadurch bestimmt, inwiefern die Schnittstellenprotokolle neu entwickelt werden müssen. Bezüglich der Anzahl der beteiligten Organisationen werden das erwartete Niveau der Zusammenarbeit zwischen den Organisationen sowie die Konflikte zwischen den Zielen der Entwicklung des SoS und denen der einzelnen Organisationen als relevante Einflussfaktoren vermutet.⁴⁴⁷

Aufbauend auf diesem Verständnis des Umfangs eines SoS werden auch im Early-Design Model verschiedene weitere Einflussfaktoren in die Prognose des Entwicklungsaufwands einbezogen. Diese wurden teilweise aus dem Post-Architecture Model übernommen und um neue oder modifizierte Einflussfaktoren ergänzt. Sie werden in Tabelle 3-6 dargestellt.⁴⁴⁸

⁴⁴⁵ Vgl. Lane /System-of-Systems/ 4; Lane /System-of-System Integration/ 1-2.

⁴⁴⁶ Vgl. Lane /System-of-Systems/ 4f.

⁴⁴⁷ Vgl. Lane /System-of-Systems/ 6f.

⁴⁴⁸ Vgl. Lane /System-of-Systems/ 7; Lane /System-of-System Integration/ 1-2.

Einflussfaktor	Erläuterung
Reifegrad der Architektur des SoS	Die Bestimmung des Reifegrades der Architektur des SoS erfolgt anhand des Detaillierungsgrades der Schnittstellenprotokolle sowie des Verständnisses des Leistungsverhaltens der Protokolle innerhalb des SoS.
Zulässige Entwicklungsdauer und -kosten	Dieser Faktor bezeichnet das Ausmaß an betriebswirtschaftlichem oder politischem Druck, die Dauer oder die Kosten der Entwicklung des SoS zu senken.
Klärung von Integrationsrisiken	Entspricht dem bereits erläuterten Faktor im Post-Architecture Model.
Reifegrad und Stabilität der Subsysteme des SoS	Auch dieser Faktor wird als Kombination verschiedener Sachverhalte verstanden. Im Einzelnen repräsentiert er das Verhältnis der Anzahl neu entwickelter Subsysteme gegenüber der Anzahl bereits außerhalb des SoS eingesetzter Subsysteme, die allgemeine Kompatibilität der Subsysteme des SoS zueinander und zu den Schnittstellenprotokollen, das Ausmaß der Veränderung der Zusammensetzung der in das SoS einzubeziehenden Subsysteme gemeinsam mit dem Ausmaß der Veränderung der Architektur des SoS sowie die Menge der Veränderungen an den Subsystemen selbst, die während der Entwicklung des SoS eintritt.
Bereitschaft der Subsysteme für die Integration	Entspricht dem bereits erläuterten Faktor im Post-Architecture Model.
Fähigkeiten des Integrationsteams	Entspricht dem bereits erläuterten Faktor im Post-Architecture Model.
Reifegrad des Integrationsprozesses	Entspricht dem bereits erläuterten Faktor im Post-Architecture Model.

Tabelle 3-6: Einflussfaktoren auf den Aufwand für die Entwicklung von SoS gemäß des Early-Design Model des COSOSIMO-Verfahrens⁴⁴⁹

3.4 Erkenntnisse zum integrationsorientierten Reengineering von Altsystemen

Der Bedarf, die Anwendungssysteme eines Unternehmens zu integrieren, besteht nicht erst, seitdem die modernen Konzepte und Technologien der Anwendungsintegration zunehmend an Verbreitung gewinnen. In der deutschsprachigen Wirtschaftsinformatik wird seit Anfang der 1990er Jahre in verschiedenen Beiträgen der Ansatz des integrationsorientierten Reengineering behandelt.⁴⁵⁰

Integrationsorientiertes Reengineering wird definiert als „die methodische Analyse eines bereits vorhandenen Softwaresystems, die eine Anhebung der vorliegenden Dokumente (Code/Dokumentation, Entwurf, Spezifikation) auf ein abstrakteres Niveau (Reverse Engineering) einschließen kann, gefolgt von der ebenfalls methodengestützten Modifikation (Forward Engineering), mit dem Ziel, das System in eine Anwendungsar-

⁴⁴⁹ Vgl. Lane /System-of-Systems/ 7; Lane /System-of-System Integration/ 1-2.

⁴⁵⁰ Vgl. beispielsweise Jung /Integrationsorientiertes Reengineering/; Eicker, Schnieder /Integrationsorientiertes Reengineering/; Eicker, Jung /Reengineering-Projekte/; Eicker u. a. /Integrationsorientiertes Reengineering/; Jung u. a. /Reengineering-Projekte/.

chitektur einbinden zu können, die insbesondere die Daten der Teilsysteme integriert.“⁴⁵¹

Dieser Ansatz bezieht sich damit auf die Daten der Anwendungssysteme als Integrationsgegenstand. Ferner wird das integrationsorientierte Reengineering insbesondere im Zusammenhang mit Altsystemen behandelt, die untereinander oder mit modernen Anwendungssystemen verknüpft werden sollen.⁴⁵² Das Ergebnis des integrationsorientierten Reengineering wird als integriertes verteiltes Anwendungssystem (IVAS) bezeichnet, definiert als „ein Anwendungssystem, das aus einem oder mehreren bereits miteinander integrierten, eventuell auf verschiedenen Netzknoten angesiedelten Anwendungssystemen besteht und dessen Architektur und Dokumentation darauf ausgerichtet sind, die Integration weiterer Anwendungssysteme zu ermöglichen.“⁴⁵³

In einem Arbeitsbericht des Instituts für Wirtschaftsinformatik der Universität Münster stellt Jung verschiedene Überlegungen dar, welche Faktoren den Aufwand des integrationsorientierten Reengineering beeinflussen. Dabei kann er sich jedoch kaum auf widerspruchsfreie empirische Erkenntnisse beziehen.⁴⁵⁴

Jung differenziert vier Teilaufgaben, deren Erfüllung mit einem spezifischen Aufwand verbunden wird: die Einarbeitung in das betroffene Altsystem (Einarbeitungsaufwand), die Änderung des betroffenen Altsystems und seine Integration in die Integrationslösung (Änderungsaufwand, Integrationsaufwand), das anschließende Testen der Integrationslösung (Testaufwand) sowie die Aktualisierung der zugehörigen Dokumentation (Dokumentationsaufwand). Darauf aufbauend werden verschiedene Aspekte erläutert, die als maßgebliche Einflussfaktoren auf den Aufwand für die Erfüllung der genannten Teilaufgaben angesehen werden. Sie beziehen sich auf die Beschaffenheit des Altsystems, die Ziele und Merkmale des Reengineering-Prozesses und die Beschaffenheit des IVAS.⁴⁵⁵ Sie werden in Tabelle 3-7 erläutert.

⁴⁵¹ Jung /Integrationsorientiertes Reengineering/ 4.

⁴⁵² Vgl. Eicker u. a. /Integrationsorientiertes Reengineering/ 138.

⁴⁵³ Jung /Integrationsorientiertes Reengineering/ 5.

⁴⁵⁴ Vgl. Jung /Integrationsorientiertes Reengineering/ 16-25; Jung u. a. /Reengineering-Projekte/ 182.

⁴⁵⁵ Vgl. Jung /Integrationsorientiertes Reengineering/ 16-25.

Einflussfaktor	Erläuterung
Auf die Beschaffenheit des Altsystems bezogene Einflussfaktoren	
Verständlichkeit des Quellcodes	<p>Die Verständlichkeit des Quellcodes wird als Resultat verschiedener, teilweise interdependenter Merkmale des Quellcodes verstanden. Dies sind im Einzelnen:</p> <ul style="list-style-type: none"> – Die Modularisiertheit, d. h. inwiefern ein Programm in Unterprogramme (Module) zerlegt ist, die unabhängig zur Erfüllung einzelner Aufgaben dienen. – Die Modul-Vollständigkeit (module completeness), d. h. inwiefern ein Modul ohne Kenntnis anderer Module verständlich ist. – Die Parametrisiertheit gemeinsam mit der Ausprägung der Datenkommunikation, d. h. auf welche Weise Daten zwischen den einzelnen Unterprogrammen weitergegeben werden und welche Interdependenzen zwischen den Unterprogrammen daraus resultieren. – Die Datenstrukturiertheit, welche die Verwendung von Typen, die Strukturierung von Daten mit Hilfe von Records und Arrays sowie die Verwendung „sprechender“ Bezeichner betrifft. – Die Programmkomplexität, welche aus der Anzahl der Elemente des Programmsystems und den zwischen ihnen bestehenden Beziehungen resultiert. „Im Großen“ kann die Komplexität auf die einzelnen Module als Elemente bezogen werden. „Im Kleinen“ kann sie dagegen auf die Struktur des Quellcodes und den Kontrollfluss bezogen werden (beispielsweise im Hinblick auf Steuerkonstrukte und Sprunganweisungen). – Die Selbstdokumentiertheit, d. h. inwiefern die Bedeutung der Bezeichner, Anweisungen und Kommentare ersichtlich ist, ohne dass zusätzliche Erläuterungen erforderlich sind. – Die verwendete Programmiersprache, da diese grundlegenden Einfluss auf die Ausprägung der zuvor genannten Merkmale des Quellcodes haben kann.
Datenunabhängigkeit	Die Datenunabhängigkeit bezeichnet das Ausmaß, in dem die physische Datenorganisation unabhängig von der logischen Datenorganisation ist und die Datenspeicherung technisch unabhängig von dem erzeugenden oder benutzenden Softwaresystem erfolgt.
Wartungszustand	Dieser Einflussfaktor bezeichnet, inwiefern das betroffene Altsystem bereits im Zuge von Wartungsaktivitäten verändert wurde, ob diese Änderungen in geeigneter Weise dokumentiert wurden und ob durch die Änderungen die Homogenität bzw. der Ordnungsgrad des Altsystems vermindert wurde.
Verfügbarkeit und Qualität der Software-Dokumentation	Dieser Einflussfaktor bezeichnet die Verfügbarkeit und Qualität der Entwickler-, Benutzer- und Betreiberdokumentation des betroffenen Altsystems. Insbesondere den Dokumenten, die während der Entwicklung des Altsystems im Rahmen der Analyse, des Entwurfs oder dem Testen erstellt wurden, wird große Bedeutung beigemessen. Die Qualität der Dokumentation wird im Sinne von Verständlichkeit, Vollständigkeit, Zweckmäßigkeit, Fehlerfreiheit, Aktualität und maschineller Weiterverarbeitbarkeit verstanden.
Verwendung moderner Entwicklungsmethoden	Dieser Einflussfaktor bezeichnet das Ausmaß, in dem moderne Methoden bei der Entwicklung des Altsystems angewandt wurden. Diese sollten die zuvor genannten Merkmale des Altsystems positiv beeinflussen. Konkrete moderne Entwicklungsmethoden werden jedoch nicht genannt.
Restnutzungsdauer der Hardware- und Softwareplattform	Dieser Einflussfaktor bezeichnet, inwiefern im Rahmen des integrationsorientierten Reengineering zusätzlich eine Portierung des Altsystems auf eine neue Hardware- oder Softwareplattform durchgeführt werden muss.

Einflussfaktor	Erläuterung
Auf die Ziele und Merkmale des Reengineering-Prozesses bezogene Einflussfaktoren	
Umfang des Zielsystems	Dieser Einflussfaktor bezeichnet den Umfang des verfolgten Zielsystems, insbesondere im Hinblick auf die Ziele, die über die Integration des Altsystems in das IVAS hinausgehend eine Änderung des Altsystems erforderlich machen (z. B. die Entwicklung einer neuen Benutzeroberfläche, die Entwicklung zusätzlicher Funktionen).
Qualifikation, Erfahrung und Motivation der Mitarbeiter	Dieser Einflussfaktor bezeichnet die Qualifikation, Erfahrung und Motivation der Mitarbeiter, welche das integrationsorientierte Reengineering durchführen. Darüber hinaus wird insbesondere im Falle einer geringen Verständlichkeit des Quellcodes das Ausmaß der geistigen Flexibilität der Mitarbeiter als relevant erachtet.
Verfügbarkeit der Entwickler des Altsystems	Dieser Einflussfaktor bezeichnet, inwiefern die an der Entwicklung des Altsystems beteiligten Personen für das integrationsorientierte Reengineering zur Verfügung stehen.
Verfügbarkeit von Entwicklungswerkzeugen	Dieser Einflussfaktor bezeichnet die Verfügbarkeit von Entwicklungswerkzeugen, welche die Durchführung der differenzierten Teilaufgaben unterstützen. In diesem Sinne kann der Einflussfaktor weiter untergliedert werden: <ul style="list-style-type: none"> – die Verfügbarkeit von Wartungswerkzeugen, – die Verfügbarkeit von Werkzeugen zur Unterstützung des Daten-Reengineering, – die Verfügbarkeit von Integrationswerkzeugen, – die Verfügbarkeit von Testwerkzeugen.
Auf die Beschaffenheit des IVAS bezogene Einflussfaktoren	
Größe des IVAS	Dieser Einflussfaktor wird angesichts der informationsorientierten Integrationsperspektive, welche dem Ansatz zugrunde liegt, auf die Anzahl der Datenobjekte im IVAS bezogen, in das das betroffene Altsystem eingebunden werden soll. Er bezeichnet damit den Umfang des IVAS-Datenmodells, mit dem das Datenmodell des zu integrierenden Altsystems abgeglichen werden muss.

Tabelle 3-7: Einflussfaktoren auf den Aufwand für das integrationsorientierte Reengineering von Altsystemen gemäß Jung⁴⁵⁶

Es wurde bereits darauf hingewiesen, dass die von Jung angeführten Überlegungen zu den Einflussfaktoren auf den Aufwand für das integrationsorientierte Reengineering von Altsystemen zunächst kaum auf widerspruchsfreien empirischen Erkenntnissen basieren. Dieser Mangel sollte zumindest teilweise durch eine 1994 durchgeführte empirische Studie behoben werden, die im Weiteren zusammenfassend dargestellt wird.⁴⁵⁷

Es wurden drei auf die Beschaffenheit des Altsystems bezogene Einflussfaktoren ausgewählt, die mit der Verständlichkeit der Datenstrukturen bzw. mit dem Datenfluss innerhalb des Altsystems in Verbindung stehen, um deren Wirkung auf den Einarbei-

⁴⁵⁶ Vgl. Jung /Integrationsorientiertes Reengineering/ 16-25.

⁴⁵⁷ Eine detaillierte Darstellung der Studie erfolgt in Jung u. a. /Reengineering-Projekte/ 181-189.

tungsaufwand in einem Labor-Experiment zu untersuchen. Im Einzelnen sind dies die Dokumentiertheit der Datenstrukturen, die Komplexität des Quellcodes und die Verfügbarkeit von Tages- bzw. Testdaten.⁴⁵⁸ Während die ersten beiden Einflussfaktoren zumindest in ähnlicher Weise in dem oben dargestellten Modell von Jung enthalten sind (Selbstdokumentiertheit und Programmkomplexität als Aspekte der Verständlichkeit des Quellcodes), stellt die Verfügbarkeit von Tages- bzw. Testdaten eine Erweiterung des Modells dar.⁴⁵⁹

Es wurde vermutet, dass sowohl ein zunehmendes Ausmaß der Dokumentiertheit der Daten als auch die Verfügbarkeit zunehmender Mengen an Tages- bzw. Testdaten zu einem abnehmenden Einarbeitungsaufwand führen würde. Dagegen wurde angenommen, dass eine zunehmende Komplexität des Quellcodes in einem zunehmenden Einarbeitungsaufwand resultieren würde.⁴⁶⁰

Für das Experiment wurde eine in der Programmiersprache Cobol entwickelte Software eines großen Kreditinstituts ausgewählt, dessen Umfang mit 1260 Programmzeilen angegeben wird. Neben der ursprünglichen Version der Software wurden verschiedene weitere Versionen konstruiert, in denen die Ausprägung der untersuchten Einflussfaktoren systematisch verändert wurde. Darauf hin sollten 45 Versuchspersonen unabhängig von einander die Datenelemente der File Section der Software nachdokumentieren und Redundanzen zwischen Datenelementen unterschiedlicher Dateien erkennen. Der damit verbundene Aufwand wurde durch die benötigte Arbeitszeit der Versuchspersonen erfasst. Die Bearbeitung der Aufgabe galt als beendet, wenn die Versuchsperson nach subjektiver Einschätzung davon ausging, dass sie keine Verbesserung der Lösung mehr erzielen könne. Ergänzend zur Bearbeitungszeit wurde auch die Vollständigkeit des Arbeitsergebnisses der Versuchspersonen beurteilt.⁴⁶¹

Anhand der Auswertung der in dem Experiment gewonnenen Daten konnte jedoch keine der aufgestellten Hypothesen zu den Einflussfaktoren bestätigt werden. Es wurde kein signifikanter Einfluss der Dokumentiertheit der Datenstrukturen auf den Entwick-

⁴⁵⁸ Vgl. Jung u. a. /Reengineering-Projekte/ 182.

⁴⁵⁹ Tagesdaten werden definiert als eine Menge von Datensätzen, die ohne Anwendung festgelegter Kriterien aus einem Datenbestand entnommen werden. Unter Testdaten wird dagegen eine Menge von Datensätzen verstanden, die konstruiert werden, um die Ausführung der Software unter Anwendung eines bestimmten Testverfahrens zu analysieren (vgl. Jung u. a. /Reengineering-Projekte/ 189).

⁴⁶⁰ Vgl. Jung u. a. /Reengineering-Projekte/ 182.

⁴⁶¹ Vgl. Jung u. a. /Reengineering-Projekte/ 183-185.

lungsaufwand festgestellt. Die Vollständigkeit des Arbeitsergebnisses stieg jedoch mit einem zunehmenden Ausmaß der Dokumentiertheit. Im Hinblick auf die Komplexität des Quellcodes konnte weder auf den Aufwand noch auf die Vollständigkeit des Arbeitsergebnisses eine signifikante Auswirkung nachgewiesen werden. Die Verfügbarkeit einer zunehmenden Menge an Tages- bzw. Testdaten schließlich resultierte widerwartend in einem zunehmenden Aufwand, während in Bezug auf die Vollständigkeit des Arbeitsergebnisses keine signifikanten Unterschiede auftraten.⁴⁶²

Ob die untersuchten Merkmalen der Beschaffenheit des Altsystems somit nicht in erwarteter Weise den Einarbeitsaufwand im Zuge des integrationsorientierten Reengineering beeinflussen oder ob die erzielten Ergebnisse lediglich auf die gewählte Operationalisierung und Messung der Variablen und die sonstige Gestaltung des Laborexperiments zurückzuführen sind, bleibt im Unklaren.⁴⁶³

3.5 Zusammenfassende Beurteilung der Erkenntnisse

Zusammenfassend scheinen die vorgestellten Untersuchungsergebnisse und Verfahren zur Aufwandschätzung den im konzeptionellen Bezugsrahmen erläuterten Charakteristika der Anwendungsintegration nicht gerecht zu werden. Dies gilt insbesondere für die Erkenntnisse zur Neuentwicklung von Anwendungssystemen, in denen sich die Spezifika der Integrationsprobleme und Integrationslösungen ebenso wenig widerspiegeln, wie die erläuterten Besonderheiten der Integrationsvorhaben und die mit ihnen verbundenen Erfolgsfaktoren und Risiken. Zwar existieren mit den vorgestellten Beiträgen zum Einsatz von Standardsoftwarekomponenten, zur Entwicklung eines SoS oder zum integrationsorientierten Reengineering von Altsystemen erste Ansätze, die verschiedenen Besonderheiten gegenüber der Neuentwicklung von Anwendungssystemen Rechnung tragen. Jedoch können auch sie angesichts der ihnen zugrunde liegenden Untersuchungsgegenstände, Begriffsverständnisse und Annahmen nur eingeschränkt auf die Anwendungsintegration im Sinne dieser Arbeit übertragen werden.

Auffällig ist das sowohl bei der Neuentwicklung von Anwendungssystemen als auch bei den COCOTS- und COSOSIMO-Verfahren vorherrschende Verständnis, die zu bewältigende Arbeitslast werde im Grundsatz durch den Umfang der zu entwickelnden

⁴⁶² Vgl. Jung u. a. /Reengineering-Projekte/ 188f.

⁴⁶³ Vgl. Jung u. a. /Reengineering-Projekte/ 188f.

Software determiniert, der wiederum in der Anzahl der Zeilen oder logischen Anweisungen des Programmcodes zum Ausdruck komme oder in Function Points gemessen werden könne.⁴⁶⁴ Diese Vorstellung scheint mit den erläuterten Charakteristika der Anwendungsintegration jedoch nicht vereinbar zu sein, gemäß denen die Auswahl und Konfiguration von Integrationsprodukten sowie deren Verknüpfung mit den zu integrierenden Anwendungssystemen gegenüber der Neuentwicklung von Softwarekomponenten an Bedeutung gewinnt. Die Schwierigkeit der an den Verbindungspunkten erforderlichen Programmierung dürfte kaum auf der Anzahl der erstellten Programmzeilen oder der logischen Anweisungen beruhen. Generell erscheint fraglich, inwiefern die traditionell mit dem größten Aufwand in Verbindung gebrachte Tätigkeit der Programmierung auch bei der Entwicklung und Bereitstellung von Integrationslösungen eine zentrale Rolle einnimmt.

Des Weiteren wird deutlich, dass bei den vorgestellten Untersuchungen und Verfahren aus wissenschaftlicher Sicht kaum von gesicherten Erkenntnissen zu den Einflussfaktoren auf den Entwicklungsaufwand gesprochen werden kann. Die postulierten Anforderungen an die Entwicklung valider und zuverlässiger Maße und zur empirischen Untersuchung von Wirkungszusammenhängen scheinen in vielen Fällen nicht erfüllt.⁴⁶⁵ Die intersubjektive Nachvollziehbarkeit der Untersuchungen ebenso wie die Verifizierung und Repräsentativität der Erkenntnisse sind oftmals nicht in angemessener Weise sichergestellt.⁴⁶⁶

Angesichts dieser Bestandsaufnahme wird die der Arbeit zugrunde gelegte Problemstellung deutlich: Es mangelt an geeigneten Methoden zur Schätzung des Entwicklungsaufwands der Anwendungsintegration. Es ist unklar, welche Faktoren in die Aufwandschätzung der Anwendungsintegration einbezogen werden müssen, um zuverlässige Schätzergebnisse zu erzielen. Schließlich fehlt es an angemessenen empirischen oder theoretischen Erkenntnissen, auf deren Grundlage Schlüsse auf die maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration gezogen werden könnten.

⁴⁶⁴ Vgl. Francalanci /Implementation Effort/ 33.

⁴⁶⁵ Vgl. Kapitel 1.1.3.

⁴⁶⁶ Vgl. diesbezüglich auch Stevenson /Software Engineering Productivity/ 6f.

4 Konzeption der Untersuchung

Ausgehend von dem konzeptionellen Bezugsrahmen und dem dargestellten Stand der Erkenntnis zu den Einflussfaktoren auf den Entwicklungsaufwand wird im Weiteren die Konzeption der Untersuchung erläutert, aufgrund derer ein Beitrag zur Lösung der aufgezeigten Problemstellung geleistet werden soll.

Dazu werden zunächst die Untersuchungsziele definiert, an denen sich die weitere Konzeption der Untersuchung ausrichtet. Anschließend werden die theoretischen Grundlagen der gewählten Untersuchungsmethode erklärt, bevor die Methode selbst detailliert spezifiziert wird. Schließlich wird der Ablauf der Untersuchung ausführlich beschrieben, um auf diese Weise die Transparenz und Nachvollziehbarkeit des Untersuchungsprozesses sicherzustellen.

4.1 Definition der Untersuchungsziele

Den Ausgangspunkt für die zweckdienliche und konsistente Konzeption einer Untersuchung bildet die Definition der Untersuchungsziele. Diese orientieren sich an dem postulierten Erkenntnisziel der Arbeit: Das Verständnis erfahrener Praktiker von den maßgeblichen Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand soll in systematischer Weise erhoben werden. Dabei soll insbesondere ein Einblick in ihre Interpretationsmuster gewonnen werden, um zu verstehen, welche Schlussfolgerungen sie bei der Schätzung des Entwicklungsaufwands ziehen.

Gemäß Briand, Morasca und Basili können fünf Zieldimensionen einer Untersuchung unterschieden werden: das Untersuchungsobjekt, der Fokus der Untersuchung, der Zweck der Untersuchung, die eingenommene Betrachtungsperspektive und das Umfeld, auf das sich die Untersuchung bezieht.⁴⁶⁷ Diese Zieldimensionen werden im Weiteren für die vorliegende Arbeit definiert:

- Die Untersuchungsobjekte sind Integrationsvorhaben, d. h. abgegrenzte (Teil-)Projekte, die mit der Absicht durchgeführt werden, eine Integrationslösung im Sinne des zugrunde gelegten Begriffsverständnisses der Anwendungsintegration zum Beheben eines bestimmten Integrationsproblems zu entwickeln und bereitzustellen.

⁴⁶⁷ Vgl. Briand, Morasca, Basili /Measures/ 1111-1114.

- Die Untersuchung fokussiert auf den mit den Integrationsvorhaben verbundenen Entwicklungsaufwand im Sinne des definierten Begriffsverständnisses und die ihn determinierenden Einflussfaktoren.
- Der Zweck der Untersuchung besteht in der Beschreibung des Verständnisses erfahrener Praktiker von den maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand und insbesondere in der Beschreibung der von ihnen angewandten Interpretationsmuster. Auf diese Weise sollen erste Grundlagen für die Entwicklung prognostischer Modelle sowie Anhaltspunkte für die praktische Aufwandschätzung geschaffen werden. Die erfahrenen Praktiker stellen damit die Subjekte der Untersuchung dar.⁴⁶⁸
- Die Untersuchung erfolgt aus der Betrachtungsperspektive der für die Entwicklung und Bereitstellung der Integrationslösung zuständigen Personen (in Abgrenzung zur Betrachtungsperspektive der Auftraggeber der Integrationsvorhaben oder der Benutzer der zu integrierenden Anwendungssysteme).
- Schließlich bezieht sich die Untersuchung auf die Mitglieder von in Deutschland ansässigen Organisationseinheiten, die Erfahrungen in der arbeitsteiligen Durchführung von Integrationsvorhaben gesammelt haben. Weitere Einschränkungen bezüglich der Merkmale der von ihnen durchgeführten Integrationsvorhaben werden nicht festgelegt.

4.2 Theoretische Grundlagen der Untersuchung

Der Zweck der Untersuchung besteht wie erläutert in der Beschreibung des Verständnisses erfahrener Praktiker von den maßgeblichen Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand und insbesondere in der Beschreibung der von ihnen angewandten Interpretationsmuster. Für die weitere Konzeption der Untersuchung ist es zunächst erforderlich, eine Vorstellung von der Beschaffenheit des Verständnisses und der Interpretationsmuster von Praktikern zu erlangen. Darauf aufbauend kann eine geeignete Erhebungsmethode bestimmt werden.

Eine geeignete Grundlage für die Untersuchung bildet diesbezüglich die Theorie der persönlichen Konstrukte (Personal Construct Theory) nach Kelly.⁴⁶⁹ Diese sagt aus,

⁴⁶⁸ Vgl. Kitchenham u. a. /Empirical Research/ 725.

⁴⁶⁹ Vgl. Kelly /Personal Constructs/.

dass Individuen ihre Umwelt verstehen, indem sie aufgrund ihrer Erlebnisse ein komplexes System persönlicher Konstrukte entwickeln. Mit Hilfe dieser Konstrukte strukturieren sie ihre Umwelt, interpretieren sie Situationen, antizipieren sie zukünftige Ereignisse und legen sie ihr eigenes Verhalten fest. Dabei überprüfen, modifizieren und erweitern sie ständig das ihnen eigene Konstruktsystem. Diese Prozesse laufen in großen Teilen unbewusst ab.⁴⁷⁰

Das Verständnis erfahrener Praktiker von den maßgeblichen Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand kommt in diesem Sinne in ihren Konstruktsystemen zum Ausdruck. Persönliche Konstrukte repräsentieren die Art und Weise, in der ein Individuum seine Umwelt mit Bedeutung versieht. Die von den Praktikern angewandten Interpretationsmuster können demnach unmittelbar auf ihre Konstruktsysteme zurückgeführt werden.

Der Begriff der persönlichen Konstrukte kann in unterschiedlicher Weise definiert werden. Fransella und Bannister zitieren diesbezüglich Kelly, ein Konstrukt sei ein Aspekt, „in dem sich zwei oder mehr Dinge gleich sind und sich damit gegenüber einem dritten oder mehr Dingen unterscheiden.“⁴⁷¹ Als zentrales Merkmal der Konstrukte wird ihre bipolare Natur angesehen, d. h. die Individuen ordnen einem Aspekt zwei entgegengesetzte Extremausprägungen zu, relativ zu denen sie Sachverhalte in ihrer Umwelt interpretieren. Die persönlichen Konstrukte eines Individuums sind somit Schemata zur gedanklichen Diskriminierung seiner Umwelt. Kelly basiert diese Festlegung auf seinen Beobachtungen, dass „wir nie etwas anerkennen, ohne gleichzeitig etwas abzuerkennen.“⁴⁷² Beispielsweise könnte ein Projektmanager ein Entwicklungsvorhaben anhand des folgenden Konstrukts beurteilen: Die zu erfüllenden Anforderungen sind vollständig verstanden – der Auftraggeber ist sich über seine Bedürfnisse selbst nicht im Klaren.

Die persönlichen Konstrukte eines Individuums bilden ein hierarchisches System, d. h. ein persönliches Konstrukt kann mehrere andere Konstrukte zusammenfassen. In diesem Sinne können einige persönliche Konstrukte als übergeordnet und andere Konstrukte als untergeordnet angesehen werden. Dabei sind nicht nur die Konstrukte, son-

⁴⁷⁰ Vgl. Tan, Hunter /Repertory Grid/ 42; Fransella, Bannister /Repertory Grid/ 3-5.

⁴⁷¹ „[A construct is] a way in which two or more things are alike and thereby different from a third or more things.“ (Kelly, zitiert nach Fransella, Bannister /Repertory Grid/ 5).

⁴⁷² “We never affirm anything without simultaneously denying something.” (Kelly, zitiert nach Fransella, Bannister /Repertory Grid/ 5). Vgl. auch Tan, Hunter /Repertory Grid/ 42.

dem auch ihre hierarchische Ordnung als persönlich anzusehen. Das bedeutet, dass die Ordnung der Konstrukte nicht unbedingt auf formaler Logik beruht, wie Pflanzen beispielsweise Bäume subsumieren, die wiederum Eichen beinhalten. Vielmehr zeigt sich die hierarchische Ordnung der persönlichen Konstrukte in ihrer Verwendung durch das Individuum.⁴⁷³

Es wird darauf hingewiesen, dass die Überordnungs- und Unterordnungsverhältnisse zwischen den Konstrukten variieren können, je nachdem aus welcher gedanklichen Perspektive das Individuum seine Umwelt interpretiert.⁴⁷⁴ Die Position eines persönlichen Konstrukts in dem hierarchischen System kann daher nur relativ beschrieben werden, d. h. ein Konstrukt kann in mehr oder weniger Fällen als mehr oder weniger übergeordnet angesehen werden.⁴⁷⁵

Individuen können in unterschiedlichem Ausmaß die persönlichen Konstrukte anderer wertschätzen oder teilen. Dies kann beispielsweise für Mitarbeiter desselben Unternehmens gelten, die einen gemeinsamen Erfahrungshintergrund haben und sich häufig austauschen.⁴⁷⁶ Die Bedeutung der einzelnen Konstrukte und ihre hierarchische Ordnung können jedoch zwischen den einzelnen Individuen variieren.⁴⁷⁷

4.3 Spezifizierung der Untersuchungsmethode

Im Sinne der erläuterten Theorie spiegelt sich das Verständnis einer Person von ihrer Umwelt in dem System ihrer persönlichen Konstrukte wider. Dieses Konstruktsystem erklärt demnach die von der Person angewandten Interpretationsmuster. Es gilt daher eine Untersuchungsmethode zu wählen, die es ermöglicht, die persönlichen Konstrukte erfahrener Praktiker zu aufwandsrelevanten Sachverhalten explizit zu machen.

⁴⁷³ Vgl. Fransella, Bannister /Repertory Grid/ 8.

⁴⁷⁴ Die Bedeutung der gedanklichen Perspektive für die Überordnungs- und Unterordnungsverhältnisse kann beispielhaft an dem Konstrukt "allein – gemeinsam" erörtert werden. Je nachdem in welchem Zusammenhang dieses Konstrukt verwendet wird, kann es aus Sicht der Person der Pol „allein“ oder der Pol „gemeinsam“ erstrebenswert sein. Davon abhängig wird sie unterschiedliche weitere Konstrukte damit verbinden, wie beispielsweise „selbständig – unselbständig“ oder „verloren – geboren“.

⁴⁷⁵ Vgl. Fransella, Bannister /Repertory Grid/ 64f.

⁴⁷⁶ Vgl. Tan, Hunter /Repertory Grid/ 42.

⁴⁷⁷ Vgl. Fransella, Bannister /Repertory Grid/ 7.

4.3.1 Repertory Grid Technique

Ein Vorgehen zur Untersuchung, wie Personen Konstrukte einsetzen, um Elemente ihrer Umwelt zu beurteilen, wird mit der Repertory Grid Technique beschrieben. Diese basiert im Wesentlichen auf drei Komponenten: den Elementen der Umwelt (engl. elements), die durch die Personen beurteilt werden, den Konstrukten (engl. constructs), anhand derer die Beurteilung der Elemente erfolgt, und den Verbindungen zwischen Elementen und Konstrukten (engl. links), d. h. der jeweiligen Einordnung der Elemente durch die Personen relativ zu den beiden Polen der Konstrukte anhand quantitativer Skalen.⁴⁷⁸

Der wissenschaftliche Einsatz der Repertory Grid Technique erfolgt im Allgemeinen in Form von strukturierten Interviews, d. h. einzelne Auskunftspersonen, deren Verständnis ihrer Umwelt erfasst werden soll, werden durch einen Interviewer in einem persönlichen Gespräch befragt. Darüber hinaus werden Anwendungsfälle der Repertory Grid Technique beschrieben, in denen mehrere Auskunftspersonen gemeinsam befragt werden oder die Methode unabhängig von einem Interviewer zum Zweck der Selbsterkenntnis angewandt wird.⁴⁷⁹ Auf solche Spezialfälle wird im Weiteren jedoch nicht eingegangen.

Ausgehend von diesen grundlegenden Charakteristika kann die Repertory Grid Technique im Sinne unterschiedlicher Forschungsansätze eingesetzt und dazu in unterschiedlicher Weise ausgestaltet werden. In diesem Zusammenhang können zum einen qualitative von quantitativen Ansätzen und zum anderen idiographische von normativen Ansätzen unterschieden werden.⁴⁸⁰

Für einen qualitativen Forschungsansatz steht die Identifizierung von Leitgedanken (engl. emerging themes) im Verständnis der Auskunftspersonen im Mittelpunkt. In diesem Sinne werden insbesondere die Erhebung von persönlichen Konstrukten der Auskunftspersonen und deren inhaltliche Analyse angestrebt. Die Verbindungen zwischen Elementen und Konstrukten anhand quantitativer Skalen sind dagegen von geringerer Bedeutung.

⁴⁷⁸ Vgl. Tan, Hunter /Repertory Grid/ 43.

⁴⁷⁹ Vgl. beispielsweise Stewart, Stewart, Fonda /Repertory Grid/ 192-208.

⁴⁸⁰ Vgl. Tan, Hunter /Repertory Grid/ 51f.

Gegenteilig dazu verhält es sich beim quantitativen Forschungsansatz. Dieser strebt nicht die inhaltliche Analyse des Verständnisses der Auskunftspersonen an, sondern er bezieht sich auf die mathematische oder statistische Analyse der Verbindungen zwischen den Elementen und den Konstrukten. Darüber sollen beispielsweise die Struktur der Konstruktsysteme der Auskunftspersonen untersucht oder Vergleiche zwischen verschiedenen Gruppen von Auskunftspersonen durchgeführt werden.⁴⁸¹

Ein idiographischer Forschungsansatz zielt auf die Beschreibung der subjektiven Erfahrungen der Auskunftspersonen in ihren eigenen Worten. Es ist daher nicht erforderlich, dass alle Auskunftspersonen die gleichen Elemente oder Konstrukte erörtern. Demgegenüber versucht ein normativer Forschungsansatz (engl. nomothetic investigation) über den Vergleich der Repertory Grids der verschiedenen Auskunftspersonen auf Gesetzmäßigkeiten ihrer Interpretationsmuster zu schließen. Um die Vergleichbarkeit der Repertory Grids zu gewährleisten, ist es dabei erforderlich, dass die Auskunftspersonen die gleichen Elemente und Konstrukte erörtern.⁴⁸²

Ausgehend vom postulierten Erkenntnisziel dieser Arbeit wird ein qualitativer und idiographischer Forschungsansatz verfolgt. Es sollen die persönlichen Konstrukte erfahrener Praktiker erhoben werden, anhand derer sie Integrationsvorhaben im Hinblick auf den damit verbundenen Entwicklungsaufwand interpretieren. Ihre subjektiven Erfahrungen sollen in ihren eigenen Worten erfasst und inhaltlich analysiert werden.

Die konzipierte Untersuchung gleicht in ihrer Ausrichtung damit einer von Moynihan beschriebenen Studie. Diese untersucht, welche situativen Faktoren von Projektleitern bei der Planung von Softwareentwicklungsvorhaben berücksichtigt werden.⁴⁸³

4.3.2 Ausgestaltung der Methode zur Erhebung persönlicher Konstrukte

4.3.2.1 Zusammenfassende Darstellung der Methode

Im Rahmen der Repertory Grid Technique wird ein Vorgehen zur Erhebung persönlicher Konstrukte beschrieben. Dieses basiert darauf, dass eine Auskunftsperson Elemente ihrer Umwelt miteinander vergleicht, insbesondere um Unterschiede zu identifizieren, die ihrer Ansicht nach bestehen. Die identifizierten Unterschiede zwischen den Elemen-

⁴⁸¹ Vgl. Tan, Hunter /Repertory Grid/ 51.

⁴⁸² Vgl. Tan, Hunter /Repertory Grid/ 51f.

⁴⁸³ Vgl. Moynihan /Project Risk/

ten werden jeweils durch zwei Formulierungen bezeichnet, die als die entgegengesetzten Pole eines persönlichen Konstrukts der Auskunftsperson interpretiert werden.⁴⁸⁴

Beispielsweise wurden in der von Moynihan beschriebenen Studie Projektleiter verschiedener Softwareunternehmen als Auskunftspersonen gewonnen. Die in den Interviews betrachteten Elemente wurden jeweils durch Entwicklungsvorhaben gebildet, an denen die Projektleiter mitgewirkt hatten. Indem die Projektleiter diese Entwicklungsvorhaben miteinander verglichen, identifizierten sie Unterschiede, die sie durch gegensätzliche Formulierungen bezeichneten. Auf diese Weise wurden Konstrukte erhoben, wie beispielsweise: „We will be developing the application from scratch – the application will be building on a current system (not ours).“⁴⁸⁵

Als wesentliche Komponenten der Methode können die Bestimmung der betrachteten Elemente und die Gestaltung der Vergleiche angesehen werden. Ein weiterführendes Verständnis des Konstruktsystems der Auskunftsperson kann ferner durch das sog. Laddering erreicht werden, bei dem durch weiterführende Fragen über-, neben- oder untergeordnete Konstrukte erhoben werden. Die Ausgestaltung dieser Komponenten muss im Einklang mit dem verfolgten Forschungsansatz und den definierten Untersuchungszielen erfolgen.

4.3.2.2 Bestimmung der betrachteten Elemente

Die Elemente definieren die Entitäten, aufgrund derer die Methode angewandt wird, d. h. die untersuchten Objekte.⁴⁸⁶ Ihnen kommt eine große Bedeutung zu, denn sie determinieren, welche Vergleiche die Auskunftsperson durchführen und welche Unterschiede sie darüber identifizieren kann. Die betrachteten Elemente beeinflussen damit maßgeblich, welche persönlichen Konstrukte der Auskunftsperson erhoben werden.

Es werden zwei unterschiedliche Wege beschrieben, die zu betrachtenden Elemente zu bestimmen. Zum einen können die Elemente der Auskunftsperson vorgegeben werden (engl. supplied elements). Zum anderen können die Elemente durch die Auskunftsperson selbst ausgewählt werden (engl. elicited elements). Diese Auswahl kann auf Ele-

⁴⁸⁴ Vgl. Tan, Hunter /Repertory Grid/ 45f.

⁴⁸⁵ Vgl. Moynihan /Project Risk/ 365.

⁴⁸⁶ Vgl. Tan, Hunter /Repertory Grid/ 43.

mente mit bestimmten Merkmalen ausgerichtet werden und auch die Anzahl zu bestimmender Elemente kann der Auskunftsperson vorgegeben werden.⁴⁸⁷

Entsprechend der definierten Untersuchungsziele stellen Integrationsvorhaben die Untersuchungsobjekte dieser Arbeit dar. Sie dienen damit auch als Elemente, die durch die Auskunftspersonen verglichen werden, um darüber deren Verständnis von den maßgeblichen Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand zu erheben. Da des Weiteren ein qualitativer und idiographischer Forschungsansatz verfolgt wird, sollen die Auskunftspersonen solche Integrationsvorhaben selbst auswählen, mit denen sie vertraut sind. Auf diese Weise soll eine möglichst differenzierte gedankliche Auseinandersetzung mit den Integrationsvorhaben ermöglicht werden. Es ist dagegen nicht erforderlich, dass die verschiedenen Auskunftspersonen eine einheitliche Menge von Integrationsvorhaben betrachten.

4.3.2.3 Gestaltung der Vergleiche

Die Erhebung der persönlichen Konstrukte basiert auf dem Vergleich der ausgewählten Elemente durch die Auskunftsperson. Dafür werden die Elemente separat auf Karten notiert, die entsprechend der durchgeführten Vergleiche nebeneinander gelegt werden (im Weiteren als Elementkarten bezeichnet). Auf diese Weise wird die gedankliche Auseinandersetzung der Auskunftsperson mit den Elementen optisch und haptisch unterstützt.⁴⁸⁸

Darauf aufbauend werden alternative Ansätze für die Gestaltung der Vergleiche beschrieben, die sich beispielsweise hinsichtlich der Anzahl und der Auswahl der jeweils miteinander verglichenen Elemente, die Erhebung der Pole der Konstrukte, die Vorgabe eines Bezugsrahmens für die Vergleiche und die Dokumentierung der erhobenen Konstrukte unterscheiden.

▪ **Anzahl der jeweils miteinander verglichenen Elemente**

Der vorherrschende Ansatz für den Vergleich der Elemente bezieht sich unmittelbar auf die erläuterte Definition der persönlichen Konstrukte: die Auskunftsperson soll Aspekte benennen, in denen sich zwei Elemente gleichen und gegenüber einem dritten Element

⁴⁸⁷ Vgl. Tan, Hunter /Repertory Grid/ 43-45.

⁴⁸⁸ Vgl. Fransella, Bannister /Repertory Grid/ 14f.

unterscheiden. Somit werden stets drei Elemente (Triaden von Elementen) in einen Vergleich einbezogen.⁴⁸⁹

Es wird jedoch darauf hingewiesen, dass sich dieser Ansatz im Falle komplexer Elemente oder für bestimmte Gruppen von Auskunftspersonen als eine zu komplizierte kognitive Aufgabe erweisen kann.⁴⁹⁰ Brown berichtet beispielsweise von einer Studie, in der früh vom Triaden-Vergleich abgekehrt wurde, weil er die teilnehmenden Unternehmensvertreter zu verärgern schien.⁴⁹¹ Als alternativer Ansatz wird vorgeschlagen, jeweils nur zwei Elemente in einen Vergleich einzubeziehen und Unterschiede zwischen ihnen bestimmen zu lassen (Dyaden von Elementen). Auch wenn die persönlichen Konstrukte aufgrund eines Vergleichs von drei oder mehr Elementen gebildet werden, besteht gemäß Fransella und Bannister kein Grund, warum die etablierten Konstrukte auch anhand von drei Elementen erhoben werden müssten.⁴⁹²

Da Integrationsvorhaben komplexe sozio-technische Systeme darstellen, deren gedanklicher Vergleich eine schwierige Aufgabe darstellt, sollen in der konzipierten Untersuchung jeweils nur zwei Elemente in einen Vergleich einbezogen werden.

▪ **Auswahl der jeweils miteinander verglichenen Elemente**

Auch für die Auswahl der in einen Vergleich einzubeziehenden Elemente werden unterschiedliche Ansätze vorgeschlagen. Beispielsweise können einige der möglichen Dyaden willkürlich durch den Interviewleiter ausgewählt werden, sämtliche möglichen Dyaden können systematisch gebildet werden oder die Auskunftsperson kann selbständig aus der Gesamtheit der betrachteten Elemente diejenigen auswählen, die in einem bestimmten Aspekt Gemeinsamkeiten oder Unterschiede aufweisen.⁴⁹³

In der konzipierten Untersuchung wird angestrebt, sämtliche Paarungen zwischen den jeweils bestimmten Integrationsvorhaben durch die Auskunftspersonen vergleichen zu lassen. Durch die möglichst große Anzahl unterschiedlicher Vergleiche wird erhofft,

⁴⁸⁹ Vgl. Fransella, Bannister /Repertory Grid/ 14f.

⁴⁹⁰ Vgl. Fransella, Bannister /Repertory Grid/ 16.

⁴⁹¹ Vgl. Brown /Qualitative Survey Research/ 302.

⁴⁹² Vgl. Fransella, Bannister /Repertory Grid/ 15f.

⁴⁹³ Vgl. Fransella, Bannister /Repertory Grid/ 14f.

eine möglichst große Vielfalt persönlicher Konstrukte der Auskunftspersonen zu erheben.

▪ **Erhebung der Pole der Konstrukte**

Wie erläutert, sind persönliche Konstrukte von bipolarer Natur. Um sie zu erfassen ist es daher stets erforderlich, ihre beiden Pole zu bezeichnen.

Es werden unterschiedliche Vorgehensweisen beschrieben, die beiden Pole eines Konstrukts zu erheben. Zum einen können beide Ausprägungen des betrachteten Aspekts unter Bezugnahme auf die gerade miteinander verglichenen Elemente bezeichnet werden. Zum anderen kann ein Pol unmittelbar auf die miteinander verglichenen Elemente bezogen werden, während der Gegensatz dazu unabhängig von den verglichenen Elementen bezeichnet wird, um auf diese Weise deutlichere Gegensätze zu erfassen. Dieses Vorgehen birgt jedoch die Gefahr, dass die Auskunftsperson lediglich die gebräuchlichen Gegensätze nennt, ohne die tatsächlichen Gegenpole der Konstrukte zu ergründen.

In jedem Fall wird gefordert, dass beide Pole durch die Auskunftsperson selbst bezeichnet werden, da verschiedene Personen unterschiedliche Ansichten darüber haben können, worin der Gegensatz zu einem erhobenen Pol besteht - schließlich basieren ihre Überlegungen auf ihren individuellen Konstruktsystemen.⁴⁹⁴

In der konzipierten Untersuchung soll ein Pol der Konstrukte unmittelbar auf die verglichenen Elemente bezogen werden. Anschließend soll es den Auskunftspersonen ermöglicht werden, die Gegensätze unabhängig von den jeweils verglichenen Elementen zu bezeichnen. Auf diese Weise soll das von den Auskunftspersonen angenommene Spektrum der Ausprägungen der Einflussfaktoren auf den Entwicklungsaufwand vollständig erfasst werden. Dabei sollen die Bezeichnungen beider Pole stets im Wortlaut der Auskunftspersonen erfasst werden.

▪ **Bezugsrahmen für die Vergleiche**

Ein weiteres Gestaltungsmerkmal der Vergleiche ist, inwiefern der Auskunftsperson ein Bezugsrahmen vorgegeben wird. So kann es der Auskunftsperson freigestellt werden, in welcher Hinsicht sie Unterschiede oder Gemeinsamkeiten zwischen den verglichenen Elementen bestimmen soll. Häufig ist es jedoch das Ziel einer Untersuchung, zweckbe-

⁴⁹⁴ Vgl. Fransella, Bannister /Repertory Grid/ 14, 105.

zogene persönliche Konstrukte zu erheben. Um dies zu erreichen, kann der Auskunftsperson ein Bezugsrahmen für die gedankliche Auseinandersetzung mit den Elementen vorgegeben werden.⁴⁹⁵

Durch die konzipierte Untersuchung soll das Verständnis der Auskunftspersonen von den Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand erfasst werden. Den Auskunftspersonen soll daher als Bezugsrahmen vorgegeben werden, Unterschiede zwischen den betrachteten Integrationsvorhaben zu identifizieren, die ihrer Meinung nach von Bedeutung für den jeweils erforderlichen Entwicklungsaufwand sind.

▪ **Dokumentierung der Konstrukte**

Schließlich werden unterschiedliche Vorgehensweisen beschrieben, die erhobenen Konstrukte zu dokumentieren. So können die von den Auskunftspersonen gewählten Bezeichnungen der Pole unmittelbar schriftlich notiert werden. Dies bedeutet, dass sich die Auskunftspersonen bewusst auf konkrete Formulierungen festlegen. Demgegenüber werden die Konstrukte in der von Moynihan beschriebenen Studie nachträglich aufgrund von Gesprächsaufzeichnungen extrahiert.⁴⁹⁶

In der konzipierten Untersuchung sollen die Bezeichnungen für die Pole der Konstrukte in Abstimmung mit den Auskunftspersonen unmittelbar schriftlich erfasst werden. Dazu werden sie, ebenso wie zuvor die Integrationsvorhaben, auf Karten notiert (im Weiteren als Konstruktkarten bezeichnet). Darauf hin sollen die Auskunftspersonen gebeten werden, die Pole der Konstrukte zu benennen, die ihrer Erfahrung nach mit einem höheren Entwicklungsaufwand verbunden sind. Zusätzlich zur Notation der Konstrukte auf den Konstruktkarten sollen die Interviews aufgezeichnet werden, um die angemessene Interpretation der Formulierungen im Zuge der inhaltlichen Analyse sicherstellen zu können.

⁴⁹⁵ Vgl. Stewart, Stewart, Fonda /Repertory Grid/ 20-22.

⁴⁹⁶ Vgl. Moynihan /Projekt Risks/ 360.

4.3.2.4 Weiterführendes Verständnis der Konstruktsysteme

Wie erläutert basiert die Methode zur Erhebung persönlicher Konstrukte darauf, dass eine Auskunftsperson Elemente miteinander vergleicht, insbesondere um Unterschiede zwischen ihnen zu identifizieren und zu bezeichnen. Ausgehend von den auf diese Weise erhobenen Konstrukten kann durch weiterführende Fragen versucht werden, die dahinter stehenden Annahmen und Interpretationen zu ergründen und damit weitere Konstrukte im System der persönlichen Konstrukte der Auskunftsperson zu erfassen. Dieses Vorgehen wird als „Laddering“ bezeichnet, das aufwärtsgerichtet, seitwärtsgerichtet oder abwärtsgerichtet durchgeführt werden kann.⁴⁹⁷

Das aufwärtsgerichtete Laddering bezweckt die schrittweise Erhebung von hierarchisch übergeordneten Konstrukten, die eine zunehmend fundamentale Rolle im Konstruktsystem der Auskunftsperson einnehmen. Dazu wird zunächst die wert- oder zweckbezogene Einschätzung der Auskunftsperson im Hinblick auf einen Pol eines persönlichen Konstrukts erbeten, um darauf hin zu fragen, warum die Auskunftsperson zu dieser Einschätzung gelangt ist. Die Antwort der Auskunftsperson repräsentiert ein übergeordnetes Konstrukt. Dieses kann in gleicher Weise hinterfragt und entsprechend fortgefahren werden, so dass eine Kette zunehmend abstrakter Konstrukte gebildet wird.⁴⁹⁸

Das seitwärtsgerichtete Laddering bezweckt die Erhebung nebengeordneter Konstrukte. Dazu wird die Auskunftsperson gebeten, weitere Aspekte zur Beurteilung der Elemente zu benennen, die sie auf dem gleichen Betrachtungsniveau ansiedeln würde.⁴⁹⁹

Das abwärtsgerichtete Laddering schließlich bezweckt die Untergliederung abstrakter Konstrukte in untergeordnete Konstrukte. Dazu wird hinterfragt, in welcher Weise sich Elemente aus Sicht der Auskunftsperson unterscheiden, die sie den verschiedenen Polen des betrachteten Konstrukts zuordnet.⁵⁰⁰

Die Auswahl der Konstrukte, die durch das Laddering weiter untersucht werden sollen, kann durch den Interviewer oder durch die Auskunftsperson erfolgen.⁵⁰¹

⁴⁹⁷ Vgl. Tan, Hunter /Repertory Grid/ 47; Reynolds, Gutman /Laddering/; Fransella, Bannister /Repertory Grid/ 16f.; Rugg u. a. /Laddering/; Stewart, Stewart, Fonda /Repertory Grid/ 22-27.

⁴⁹⁸ Vgl. beispielsweise Fransella, Bannister /Repertory Grid/ 16f.

⁴⁹⁹ Vgl. Rugg u. a. /Laddering/ 221.

⁵⁰⁰ Vgl. beispielsweise Stewart, Stewart, Fonda /Repertory Grid/ 23f.

⁵⁰¹ Vgl. Reynolds, Gutman /Laddering/ 15.

In der konzipierten Untersuchung wird auf die gezielte Durchführung des Laddering verzichtet, denn es wird nicht davon ausgegangen, dass auf diese Weise weitere Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand identifiziert werden können.⁵⁰² Gleichwohl sollen die Hintergründe solcher Konstrukte erfragt werden, die gemäß der Erfahrung der Auskunftspersonen besondere Relevanz für den Entwicklungsaufwand besitzen. Ebenso soll versucht werden, abstrakt erscheinende Konstrukte zu konkretisieren. Auf diese Weise sollen möglichst reichhaltige Informationen gewonnen und die Bedeutung der erhobenen Konstrukte sowie die Eignung der dokumentierten Formulierungen reflektiert werden, ohne dass die Antworten auf die Nachfragen ihrerseits als Konstrukte in die inhaltliche Analyse einfließen.

4.3.3 Eignung der Methode zur Erreichung des Erkenntnisziels der Arbeit

Die Repertory Grid Technique und insbesondere die erläuterte Methode zur Erhebung persönlicher Konstrukte haben sich bereits bei der Untersuchung verschiedener Aspekte im Kontext der Wirtschaftsinformatik bewährt.⁵⁰³ Es wird als die wesentliche Stärke der Methode angesehen, dass sie es ermöglicht, die „Bausteine der Erkenntnis“ zu erschließen, die von einzelnen Personen besessen oder von mehreren Personen geteilt werden.⁵⁰⁴ Dies bedeutet, dass zum einen individuelle Erfahrungen und Interpretationsmuster erhoben werden können und zum anderen Überschneidungen zwischen den Konstruktsystemen verschiedener Personen identifiziert werden können. Dabei ermöglicht es die Methode, relativ reichhaltige Informationen zu gewinnen, denn sie erlaubt den Auskunftspersonen, ihr Verständnis von ihrer Umwelt in eigenen Worten zu beschreiben.⁵⁰⁵

Aufgrund der intensiven Auseinandersetzung mit den Ansichten der Auskunftspersonen ermöglicht die Methode zudem, aussagekräftige Erkenntnisse auf der Grundlage eines relativ kleinen Erhebungsumfangs zu erlangen. Gemäß Tan und Hunter reichen häufig

⁵⁰² So erscheint die Methode des Laddering insbesondere geeignet, um zunehmend fundamentale Werte einer Person zu ergründen. Da in der vorliegenden Untersuchung jedoch keine Werturteile sondern das Verständnis der Praktiker von realen Phänomenen erhoben werden soll, wird nicht davon ausgegangen, dass lange und aufschlussreiche Konstruktketten gebildet werden könnten.

⁵⁰³ Vgl. beispielsweise Moynihan /Project Risk/; Tan, Hunter /Repertory Grid/; Davis /Information Systems Implementation and Internalization/; Tan /Business-IT Alignment/; Hunter, Beck /Cross-Cultural Information Systems Research/; Whyte, Bytheway /Information System Success/.

⁵⁰⁴ „Its main strength is that the technique reaches the shared and unique components of individuals' cognitive construction systems.“ (Tan, Hunter /Repertory Grid/ 53)

⁵⁰⁵ Vgl. Fransella, Bannister /Repertory Grid/ 111.

15 bis 25 Auskunftspersonen aus, um das bestehende Verständnis einer Domäne angemessen zu approximieren.⁵⁰⁶

Das Verständnis eines Individuums von seiner Umwelt kann in Teilen als explizites und in anderen Teilen als implizites Wissen (engl. tacit knowledge) angesehen werden. In diesem Sinne ist sich eine Auskunftsperson lediglich zu einem Teil ihrer Erfahrungen und Interpretationsmuster bewusst und in der Lage, sie frei zu erläutern. Der andere Teil ihres Wissens ist einer Auskunftsperson dagegen unbewusst bzw. es ist ihr nicht möglich, es losgelöst von einer konkreten Anwendung präzise zu beschreiben.⁵⁰⁷ Von dieser Kombination aus explizitem und implizitem Wissen der Auskunftspersonen ist auch in dieser Arbeit auszugehen.⁵⁰⁸

Als weitere Stärke der Methode zur Erhebung persönlicher Konstrukte kann diesbezüglich angeführt werden, dass sie über die Auseinandersetzung mit den Elementen die Erschließung beider Wissensarten ermöglicht.⁵⁰⁹ Dies bedeutet, dass die Auskunftspersonen ihre Interpretationsmuster nicht frei beschreiben müssen, sondern sie über den Vergleich konkreter Integrationsvorhaben unmittelbar anwenden und dadurch auch implizites Wissen erschließen können.

Angesichts der erläuterten Stärken der Methode zur Erhebung persönlicher Konstrukte und ihrer wiederholt erfolgreichen Anwendung im Kontext der Wirtschaftsinformatik wird sie als geeignet angesehen, das Erkenntnisziel dieser Arbeit zu erreichen.

Es muss jedoch einschränkend darauf hingewiesen werden, dass die anhand der Methode erhobenen Konstrukte nicht unmittelbar mit den persönlichen Konstrukten der Auskunftspersonen gleichgesetzt werden können. In diesem Sinne betonen Fransella und Bannister: „[...] we should never assume that a construct is the same as its verbal label. A construct is a discrimination, *not* a verbal label.“⁵¹⁰

⁵⁰⁶ Vgl. Tan, Hunter /Repertory Grid/ 50.

⁵⁰⁷ Vgl. Trittman /Wissenstransfers in der Softwareentwicklung/ 38-40. Rugg u. a. erläutern dies beispielhaft anhand des Schreibens mit einer Tastatur. Erfahrene Maschinenschreiber müssen sich die Nutzung der Tastatur im Allgemeinen bildlich vorstellen, um auf die Frage antworten zu können, welcher Buchstabe rechts neben dem Buchstaben ‚g‘ angeordnet ist (vgl. Rugg u. a. /Laddering/ 218).

⁵⁰⁸ Vgl. Jorgensen /Expert Estimation/ 37-39.

⁵⁰⁹ Vgl. Rugg u. a. /Laddering/ 218f.

⁵¹⁰ Fransella, Bannister /Repertory Grid/ 2.

Darüber hinaus ist es nicht möglich, die komplexen Konstruktsysteme der Auskunftspersonen in ihrer Gesamtheit zu erfassen. Die artikulierten Interpretationsmuster der Auskunftspersonen stellen daher lediglich das möglichst präzise Abbild eines begrenzten Ausschnitts ihrer Konstruktsysteme dar.

4.3.4 Gestaltung des Auswahlverfahren der Auskunftspersonen

Der Definition der Population (Grundgesamtheit) der Untersuchungssubjekte wird im Allgemeinen eine große Bedeutung für Gewinnung empirischer Erkenntnisse beigemessen. Kitchenham u. a. betonen in diesem Sinne, dass es nicht möglich sei, Schlüsse aus den Ergebnissen einer Untersuchung zu ziehen, ohne zu wissen, aus welcher Population die Auskunftspersonen stammen.⁵¹¹ Da eine Vollerhebung, d. h. die Befragung sämtlicher Untersuchungssubjekte der definierten Population, im Allgemeinen aus Praktikabilitäts- und Wirtschaftlichkeitsgründen zu verwerfen ist, gilt es darüber hinaus, das Verfahren festzulegen, aufgrund dessen die Auskunftspersonen ausgewählt werden.⁵¹²

▪ Population der Untersuchungssubjekte

Entsprechend den definierten Untersuchungszielen wird die für diese Arbeit relevante Population von Untersuchungssubjekten zunächst durch die Gesamtheit der Personen gebildet, die als Mitglieder von in Deutschland ansässigen Organisationseinheiten an der arbeitsteiligen Entwicklung und Bereitstellung von Integrationslösungen mitwirken. Darüber hinaus besteht der Zweck der Untersuchung in der Beschreibung des Verständnisses erfahrener Personen. In diesem Zusammenhang ist es notwendig, zu spezifizieren, wann eine Person als erfahren angesehen werden kann, d. h. welchen Erfahrungshintergrund eine Person aufweisen muss, um der Population der Untersuchungssubjekte zugezählt zu werden.

Aus Sicht der Theorie der persönlichen Konstrukte können Personen in einem bestimmten Zusammenhang als erfahren angesehen werden, wenn sie aufgrund ihrer Erlebnisse ein System persönlicher Konstrukte entwickelt haben, mit deren Hilfe sie für den betreffenden Zusammenhang erfolgreich in der Lage sind, ihre Umwelt zu strukturieren, Situationen zu interpretieren, zukünftige Ereignisse zu antizipieren und ihr eigenes Verhal-

⁵¹¹ Vgl. Kitchenham u. a. /Empirical Research/ 725.

⁵¹² Vgl. Kitchenham u. a. /Empirical Research/ 725.

ten festzulegen. Im Rahmen ihrer Erlebnisse hatten sie die Möglichkeit, kontextbezogene Konstrukte auszubilden, sie zu überprüfen, sie zu modifizieren und zu ergänzen.⁵¹³

Wie erläutert können Integrationsvorhaben als neuartiger Typ von Softwareentwicklungsvorhaben angesehen werden, der sich durch spezifische Charakteristika der zugrunde liegenden Problemstellungen, der Problemlösungen, der durchzuführenden Teilaufgaben und der organisatorischen Gestaltungsmaßnahmen auszeichnet. Insofern ist davon auszugehen, dass die erfolgreiche Durchführung von Integrationsvorhaben mit spezifischen Konstrukten der beteiligten Personen verbunden ist, d. h. spezifische Erfahrung erfordert.⁵¹⁴

In Anlehnung an MacCormack, Verganti und Iansiti erscheint es für den Aufbau von Erfahrung im neuartigen Kontext der Anwendungsintegration als wichtig, an mehreren verschiedenen Integrationsvorhaben mitgewirkt zu haben.⁵¹⁵ Dies ermöglicht die Auseinandersetzung mit unterschiedlichen Integrationsproblemen, unterschiedlichen Integrationslösungen und unterschiedlichen herrschenden Bedingungen bei deren Entwicklung und Bereitstellung. Durch die Mitarbeit in verschiedenen Integrationsvorhaben können zudem Gemeinsamkeiten und Unterschiede hinsichtlich des erforderlichen Entwicklungsaufwands festgestellt und interpretiert werden. Auf diese Weise kann ein differenziertes Konstruktsystem zu aufwandsrelevanten Sachverhalten im Kontext der Anwendungsintegration aufgebaut, überprüft und weiterentwickelt werden.

Für die konzipierte Untersuchung sollen daher Auskunftspersonen bestimmt werden, die bereits an mehreren verschiedenen Integrationsvorhaben mitgewirkt haben (in Abgrenzung zur absoluten Dauer der Mitarbeit in Integrationsvorhaben).

Des Weiteren ist davon auszugehen, dass das Verständnis einer Person von den Einflussfaktoren auf den Entwicklungsaufwand dadurch geprägt wird, welche Einblicke sie im Rahmen der durchgeführten Integrationsvorhaben in die erbrachte personelle Arbeitsleistung und die Spezifika der Integrationsprobleme, der Integrationslösungen, der

⁵¹³ Vgl. Tan, Hunter /Repertory Grid/ 42; Fransella, Bannister /Repertory Grid/ 3-5. Dieses Verständnis der Erfahrung einer Person steht damit im Einklang mit verbreiteten Konzepten zum Erfahrungslernen (vgl. Trittman /Wissenstransfers in der Softwareentwicklung/ 65).

⁵¹⁴ Es herrscht die verbreitete Ansicht, dass Erfahrung kontextgebunden ist (vgl. beispielsweise Nelson u. a. /Software Operations Support Expertise/ 477).

⁵¹⁵ Vgl. MacCormack, Verganti, Iansiti /Flexible Development Process/ 137f.

Bearbeitung der Teilaufgaben und der organisatorischen Gestaltungsmaßnahmen hatte. In der konzipierten Untersuchung sollen daher solche Personen befragt werden, die in einer Weise an den betrachteten Integrationsvorhaben beteiligt waren, die es ihnen erlaubt, den jeweils benötigten Entwicklungsaufwand in seiner Gesamtheit und ein möglichst breites Spektrum potenzieller Einflussfaktoren zu erfassen. In diesem Sinne erscheinen insbesondere Personen relevant, die leitende Rollen in Integrationsvorhaben einnehmen.⁵¹⁶

▪ **Festlegung des Auswahlverfahren**

Das Ziel der Auswahl der Auskunftspersonen besteht im Falle quantitativer oder normativer Forschungsansätze darin, eine repräsentative Teilmenge der definierten Population zu erfassen. Dabei werden insbesondere zufallsbasierte Auswahlverfahren gefordert, um über statistische Analysen der Untersuchungsergebnisse Schlüsse auf die gesamte Population ziehen zu können (statistisches Sampling).⁵¹⁷

Im Falle qualitativer oder idiographischer Forschungsansätze sind statistische Rückschlüsse jedoch von geringer Bedeutung. Vielmehr stellt sich in der vorliegenden Untersuchung die Aufgabe, einen möglichst großen Ausschnitt aus dem Verständnis der Untersuchungssubjekte zu den Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand zu erheben.⁵¹⁸ Die Festlegung bestimmter Merkmale für die Auswahl der Auskunftspersonen erscheint dabei vorab nicht möglich: der Umfang der Population der Untersuchungssubjekte ist nicht bekannt, zudem ist nicht mit Gewissheit abzusehen, welche Merkmale der Untersuchungssubjekte für deren Verständnis von den Einflussfaktoren relevant und wie diese in der Population verteilt sind. Anstelle einer Vorab-Festlegung der Struktur der Stichprobe erscheint daher eine schrittweise Festlegung der Struktur im Untersuchungsprozess als angemessen.

Diese Strategie der schrittweisen Auswahl orientiert sich am Konzept des theoretischen Sampling. Dieses bezeichnet gemäß Glaser und Strauss „den Prozess der Datensammlung zur Generierung von Theorien, wobei der Forscher seine Daten gleichzeitig sammelt, kodiert und analysiert und dabei entscheidet, welche Daten als nächste gesammelt

⁵¹⁶ Bezug nehmend auf das vorgestellte Rollenmodell nach Yee und Apte erscheinen diesbezüglich insbesondere Projektleiter und Chefarchitekten relevant (vgl. Kapitel 2.3.2).

⁵¹⁷ Vgl. Kitchenham u. a. /Empirical Research/ 725.

⁵¹⁸ Vgl. Nelson u. a. /Software Operations Support Expertise/ 483.

werden sollten und wo sie zu finden sind, um seine Theorie zu entwickeln, während sie emergiert. Dieser Prozess der Datensammlung wird durch die emergierende Theorie kontrolliert.“⁵¹⁹ Flick beschreibt die schrittweise Auswahl als allgemeines Prinzip qualitativer Forschung, bei der die Suche nach und Analyse von abweichenden Fällen zur Theorieentwicklung im Mittelpunkt steht.⁵²⁰

Theoretisches Sampling	Statistisches Sampling
Der Umfang der Grundgesamtheit ist vorab unbekannt.	Der Umfang der Grundgesamtheit ist bekannt.
Die Merkmale der Grundgesamtheit sind nicht vorab bekannt.	Die Merkmalsverteilung in der Grundgesamtheit ist abschätzbar.
Es erfolgt eine mehrmalige Ziehung von Stichprobenelementen nach jeweils neu festgelegten Kriterien.	Es wird eine einmalige Ziehung einer Stichprobe nach einem vorab festgelegten Plan durchgeführt.
Die Stichprobengröße ist vorab nicht definiert.	Die Stichprobengröße ist vorab definiert.
Das Sampling wird beendet, wenn eine theoretische Sättigung erreicht ist.	Das Sampling ist beendet, wenn die gesamte Stichprobe untersucht ist.

Tabelle 4-1: Theoretisches versus statistisches Sampling⁵²¹

Dem Streben nach einer generellen Verallgemeinerbarkeit der Untersuchungsergebnisse wird in dieser Arbeit somit die möglichst reichhaltige Beschreibung subjektiver Erfahrungen entgegengestellt. Wenn die Entwicklung des Konstruktsystems einer Person auf deren Erlebnissen basiert, erscheint es für die Untersuchung daher erstrebenswert, Auskunftspersonen mit vielfältigen Erlebnishintergründen im Kontext der Anwendungsintegration zu befragen. Es sollen daher Mitarbeiter verschiedener Organisationseinheiten in die Untersuchung einbezogen werden, die darüber hinaus an einem möglichst breiten Spektrum verschiedenartiger Integrationsvorhaben mitgewirkt haben sollen.

Den Prozess der Datensammlung durch die emergierende Theorie zu kontrollieren erscheint dabei nur eingeschränkt möglich. Für eine gezielte Auswahl von Auskunftspersonen aufgrund der erhobenen persönlichen Konstrukte vorangehender Interviewpartner mangelt es an einem angemessenen theoretischen Bezugsrahmen.

Als Indikator für eine theoretische Sättigung der Erkenntnisse nennen verschiedene Autoren, inwiefern in den zuletzt durchgeführten Gesprächen neuartige Konstrukte erho-

⁵¹⁹ Glaser und Strauss /Grounded Theory/ 45, zitiert nach Flick /Qualitative Forschung/ 82.

⁵²⁰ Vgl. Flick /Qualitative Forschung/ 85f.

⁵²¹ Übernommen aus Flick /Qualitative Forschung/ 83.

ben wurden.⁵²² Sofern über mehrere Gespräche hinweg nur solche Konstrukte erfasst wurden, die eine große inhaltliche Ähnlichkeit zu den zuvor erhobenen Konstrukten aufweisen, wird daraus geschlossen, dass mit ausreichend großer Wahrscheinlichkeit das Verständnis einer Domäne angemessen approximiert werden konnte und keine weiteren Auskunftspersonen mehr in die Untersuchung einbezogen werden müssen. Gemäß Tan und Hunter ist dieser Punkt häufig nach 15 bis 25 Auskunftspersonen erreicht.⁵²³

4.3.5 Anforderungen an die Auswahl der Integrationsvorhaben

Die Erhebung der persönlichen Konstrukte basiert auf dem Vergleich von Elementen, d. h. im Falle der konzipierten Untersuchung, auf dem Vergleich von Integrationsvorhaben. Es wurde bereits darauf hingewiesen, dass den Elementen damit eine große Bedeutung für die Untersuchungsergebnisse zukommt.

Analog zu den vorangehenden Ausführungen zur Auswahl der Auskunftspersonen erscheint eine Festlegung der Gesamtheit der betrachteten Integrationsvorhaben vorab nicht möglich.⁵²⁴ Es können jedoch verschiedene Anforderungen an die Auswahl der Integrationsvorhaben im Kontext der einzelnen Interviews spezifiziert werden.

Diesbezüglich betonen verschiedene Autoren Kriterien, die bei der Auswahl der Elemente beachtet werden sollten. Beispielsweise wird gefordert, dass die miteinander verglichenen Elemente diskret und aus der Sicht der Auskunftsperson präzise definiert sind.⁵²⁵ In der konzipierten Untersuchung muss daher eine sorgfältige Abgrenzung der betrachteten Integrationsvorhaben aus Sicht der Auskunftspersonen sichergestellt werden.

Ferner wird es als wichtig erachtet, dass die Auskunftspersonen solche Integrationsvorhaben auswählen, mit deren Charakteristika sie in angemessenem Maße vertraut sind,

⁵²² Vgl. Moynihan /Project Risk/ 362; Tan, Hunter /Repertory Grid/ 50.

⁵²³ Vgl. Tan, Hunter /Repertory Grid/ 50.

⁵²⁴ Zwar wird es für die Untersuchung als wünschenswert angesehen, der Vielfalt der im konzeptionellen Bezugsrahmen erläuterten Charakteristika von Integrationsvorhaben Rechnung zu tragen, die Bedeutung bestimmter Merkmale der Integrationsvorhaben für den mit ihnen verbundenen Entwicklungsaufwand kann jedoch im Sinne des zugrunde gelegten Forschungsproblems nicht abgesehen werden.

⁵²⁵ Vgl. Tan, Hunter /Repertory Grid/ 45; Fransella, Bannister /Repertory Grid/ 13; Stewart, Stewart, Fonda /Repertory Grid/ 30-32.

um differenzierte Vergleiche durchführen zu können. Dabei wird es jedoch als unerheblich angesehen, ob die ausgewählten Integrationsvorhaben bereits längere Zeit zurückliegen oder womöglich in einem früheren Beschäftigungsverhältnis der Auskunftsperson durchgeführt wurden.⁵²⁶ Zwar ist bekannt, dass die menschliche Erinnerung an vergangene Ereignisse durch vielfältige Verzerrungen und Unvollkommenheit gekennzeichnet ist, so dass die Erörterungen weit zurück liegender Integrationsvorhaben womöglich nicht den tatsächlichen Gegebenheiten entsprechen.⁵²⁷ Dieser Effekt erscheint jedoch nicht von Relevanz, da in der konzipierten Untersuchung das aktuelle Verständnis der Auskunftspersonen von den maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand erhoben werden soll und keine präzise Dokumentation der betrachteten Integrationsvorhaben angestrebt wird.

Schließlich beeinflusst die Vielfalt der durch eine Auskunftsperson betrachteten Elemente die Wahrnehmung verschiedenartiger Unterschiede und damit die Menge der persönlichen Konstrukte, die erhoben werden können. Einzelne Autoren empfehlen diesbezüglich, jeweils vier bis fünf Elemente für die Durchführung von sechs bis zehn Vergleichen auszuwählen.⁵²⁸ In der konzipierten Untersuchung wird angestrebt, diese Anzahl verglichener Integrationsvorhaben zu erreichen. Darüber hinaus sollen die Auskunftspersonen gebeten werden, möglichst solche Integrationsvorhaben auszuwählen, die sich hinsichtlich des mit ihnen verbundenen Entwicklungsaufwands oder anderer Merkmale unterscheiden.

4.3.6 Präzisierung des Bezugsrahmens für die Vergleiche

Um die Vergleichbarkeit der erhobenen Konstrukte der verschiedenen Auskunftspersonen zu gewähren, soll eine einheitliche gedankliche Auseinandersetzung der Auskunftspersonen mit den Integrationsvorhaben erreicht werden. Einem gemeinsamen Verständnis des betrachteten Entwicklungsaufwands wird daher große Bedeutung beigemessen. Über die aufgestellte Definition des Entwicklungsaufwands hinaus werden im Folgenden weitere Präzisierungen vorgenommen.

⁵²⁶ Diese Ansicht wird beispielsweise auch in der von Moynihan beschriebenen Studie beschrieben (vgl. Moynihan /Project Risk/ 360.

⁵²⁷ Vgl. Rugg u. a. /Laddering/ 216f.

⁵²⁸ Vgl. Moynihan /Project Risk/ 364; Tan, Hunter /Repertory Grid/ 46f.

Größere Integrationsvorhaben können in Zusammenarbeit verschiedener Auftragnehmer durchgeführt werden bzw. ein Auftragnehmer kann seinerseits Teilaufgaben an andere Unternehmen vergeben. Darüber hinaus können die Auftraggeber selbst an der Erfüllung von Teilaufgaben mitwirken. Diesbezüglich soll den Auskunftspersonen nahe gelegt werden, die Integrationsvorhaben im Hinblick auf den gesamten Entwicklungsaufwand zu beurteilen, unabhängig davon, in welcher der beteiligten Organisationseinheiten er anfällt.

In der Praxis kann die Anwendungsintegration mit einer Veränderung der Strategie, der Geschäftsprozesse oder der Menge der eingesetzten Anwendungssysteme einhergehen. Diese Veränderungen erfordern ebenso wie die Entwicklung und Bereitstellung der Integrationslösung den Einsatz personeller Arbeitsleistung. Die Auskunftspersonen sollen jedoch gebeten werden, diesen Aufwand nicht in die weitere Betrachtung einzubeziehen und stattdessen von gegebenen Strategien, Geschäftsprozessen und Anwendungssystemen auszugehen. Es soll lediglich der mit der Integration der Anwendungssysteme verbundene Entwicklungsaufwand reflektiert werden.

Schließlich kann einer Auftragsvergabe für die Entwicklung und Bereitstellung einer Integrationslösung in unterschiedlichem Maße die Gewinnung eines Verständnisses des Integrationsproblems und die Konzeption der Integrationslösung voraus gehen (womöglich in einem eigenständigen Projekt). Eine Vergleichbarkeit der frühen Abschnitte der Integrationsvorhaben kann daher nicht abgesehen werden. Darüber hinaus erscheint es erstrebenswert, die in dieser Arbeit gewonnenen Erkenntnisse zu den Einflussfaktoren auf den Entwicklungsaufwand dem dargestellten bisherigen Stand der Erkenntnis gegenüber zu stellen. Um auch in diesem Fall eine möglichst hohe Vergleichbarkeit der Betrachtungsperspektive zu erreichen, sollten die Auskunftspersonen bei der Betrachtung der Integrationsvorhaben von einem Zeitpunkt ausgehen, an dem die Initialisierung der Projekte abgeschlossen, die zu entwickelnde Integrationslösung zumindest grob entworfen und eine entsprechend weitgehende Analyse bereits durchgeführt worden ist. Davon ausgehend sollte der gesamte Aufwand in die Überlegungen einbezogen werden, der für weitere Analyse- und Entwurfstätigkeiten, die Implementierung und Überleitung in die Nutzung sowie die begleitenden Unterstützungsaufgaben erbracht wurde.

4.4 Ablauf der Untersuchung

Im Weiteren werden die Vorbereitung der Interviews, die Identifikation der Untersuchungsteilnehmer und die Durchführung der Interviews beschrieben.

4.4.1 Vorbereitung der Interviews

Um die angemessene Anwendung der spezifizierten Untersuchungsmethode zur Gewinnung valider Ergebnisse sicherzustellen, wurde die Repertory Grid Technique im Juli 2005 in Gesprächen mit mehreren Mitarbeitern des Lehrstuhls für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln erprobt. Dieses Vorgehen ermöglichte es dem Interviewer, praktische Erfahrungen mit der Methode zu sammeln und die Auswirkungen einer variierenden Gestaltung der Vergleiche auf die Untersuchungsergebnisse zu ermitteln. In diesem Sinne betonen Fransella und Bannister: „The elicitation and laddering of constructs is an art and not a science. So the examiner must expect to have to gain experience in this art and so learn to minimise his influence in determining the constructs given.“⁵²⁹

Der Erprobung der Methode wurde ein spezifischer und vom Kontext dieser Arbeit abweichender Untersuchungsgegenstand zugrunde gelegt: Es sollte das Verständnis der Lehrstuhlmitarbeiter von den maßgeblichen Einflussfaktoren auf den mit Diplomarbeiten verbundenen Betreuungsaufwand erhoben werden. Unter Betreuungsaufwand wurde dabei der mengenmäßige Verbrauch an Arbeitsleistung der zuständigen Lehrstuhlmitarbeiter vom Beginn der Konzeption bis zum Abschluss der Vorkorrektur einer Diplomarbeit verstanden.

Dieser Untersuchungsgegenstand wurde gewählt, da die betreffenden Lehrstuhlmitarbeiter ebenso wie der Interviewer über mehrjährige Erfahrungen in der Betreuung von Diplomarbeiten verfügen. Der Interviewer konnte daher die Auswirkungen einer variierenden Gestaltung der Vergleiche auf die in den Gesprächen erhobenen Konstrukte vor dem Hintergrund seiner eigenen Erfahrungen reflektieren. Keiner der teilnehmenden Lehrstuhlmitarbeiter war mit der Repertory Grid Technique und der zugrunde liegenden Theorie der persönlichen Konstrukte vertraut. Der zweckbezogene Vergleich verschiedener betreuter Diplomarbeiten sollte ihnen damit ebenso neuartig erscheinen, wie den späteren Teilnehmern der Untersuchung.

⁵²⁹ Fransella, Bannister /Repertory Grid/ 108.

Durch die Erprobung der Methode wurden wertvolle Erfahrungen zur Gesprächsführung gesammelt, insbesondere in welcher Weise den Auskunftspersonen beim Vergleich der Elemente und der Bezeichnung der Pole eines Konstrukts assistiert werden kann, ohne sie dabei ungewollt in ihren Überlegungen zu beeinflussen. Diesbezüglich unterstreichen Fransella und Bannister: „Perhaps the most important rule to bear in mind when [...] eliciting constructs, is that the examiner must LISTEN. He does not have to be silent. He can mutter, nod approval, even rephrase what the person has said and ask whether this was what he meant, but he must never *impose* constructs. This is where the art lies.“⁵³⁰

4.4.2 Identifikation der Untersuchungsteilnehmer

Die Identifikation der Auskunftspersonen erfolgte wie erläutert schrittweise und parallel zur Durchführung der Interviews.

Zunächst wurde versucht, unter den Mitgliedern des im Köln-Bonner Raum tagenden Arbeitskreises „Service-orientierte Architekturen (SOA)“ im Fachbereich Wirtschaftsinformatik der Gesellschaft für Informatik e.V. geeignete Auskunftspersonen zu identifizieren. Der Arbeitskreis bezweckt die unternehmensübergreifende Diskussion verschiedener fachlicher und technischer Aspekte serviceorientierter Architekturen und setzt sich in diesem Rahmen auch mit der Anwendungsintegration auseinander.⁵³¹ Angesichts der begrenzten zeitlichen und personellen Mittel zur Durchführung der Untersuchung sprach die vermeintliche räumliche Nähe der potenziellen Interviewpartner für dieses Vorgehen im Sinne des „Convenience Sampling“.⁵³²

Da jedoch auch nach wiederholtem Aufruf zur Teilnahme an der Untersuchung nur eine geringe Anzahl von Auskunftspersonen gewonnen werden konnte, wurde zusätzlich dazu übergegangen, verschiedene Unternehmen direkt auf Mitarbeiter mit einem geeigneten Erfahrungshintergrund anzusprechen. Die Auswahl dieser Unternehmen orientierte sich insbesondere an den von ihnen ausgewiesenen Kompetenzen im Kontext der

⁵³⁰ Fransella, Bannister /Repertory Grid/ 108 (die Hervorhebungen entsprechen dem Original).

⁵³¹ Vgl. Homepage des Arbeitskreises: <http://www.soai.org/modules/cjaycontent/index.php?id=27>, Aufruf am 04.03.2006.

⁵³² Das Konzept des Convenience Sampling bezeichnet ein Vorgehen, bei dem diejenigen Auskunftspersonen in die Untersuchung einbezogen werden, die aus Sicht des Untersuchenden mit geringem Aufwand identifiziert und erreicht werden können (vgl. Flick /Qualitative Forschung/ 88).

Anwendungsintegration und ihrer Größe, gemessen an der Höhe ihres Umsatzes. Dabei wurde auf Listen der führenden Unternehmen für IT-Beratungs- und Systemintegration, IT-Services oder Business Innovation/Transformation zurückgegriffen.⁵³³ Die angesprochenen Unternehmen zeigten durchweg großes Interesse an der Untersuchung, so dass zügig weitere Teilnehmer gewonnen werden konnten.

Schließlich erwiesen sich einige der Untersuchungsteilnehmer selbst als große Unterstützung, denn sie fahndeten unter ihren Kollegen nach weiteren potenziellen Interviewpartnern und warben erfolgreich für die Teilnahme an der Untersuchung.

Mit der Identifizierung der Auskunftspersonen ging eine telefonische Kontaktaufnahme durch den Interviewer einher. In diesen Gesprächen wurde zunächst die Zielsetzung der Untersuchung dargestellt, daraufhin schilderten die potenziellen Interviewteilnehmer ihren spezifischen Erfahrungshintergrund. Sofern dieser den spezifizierten Anforderungen entsprach und die betreffenden Personen sowohl die Bereitschaft als auch die zeitliche Möglichkeit zur Teilnahme an der Untersuchung hatten, wurde ein Interviewtermin vereinbart. Darüber hinaus wurde im Rahmen des Telefonats bereits das Einverständnis der Untersuchungsteilnehmer für die akustische Aufzeichnung der Interviews eingeholt.

4.4.3 Durchführung der Interviews

Die Durchführung der Interviews mit den identifizierten Untersuchungsteilnehmern erfolgte von Mitte August 2005 bis Ende November 2005. Insgesamt wurden im Rahmen der Studie 22 Interviews mit Mitarbeitern von 10 Unternehmen geführt. Sämtliche Gespräche wurden durch denselben Interviewer anhand eines detaillierten Interviewleitfadens durchgeführt. Auf diese Weise konnte eine größtmögliche Vergleichbarkeit der Gesprächsführung sichergestellt werden.

Die Dauer der Interviews betrug jeweils 1,5 bis 2 Stunden. Sie gliederten sich in fünf Abschnitte: eine Einführung, die Vereinbarung begrifflicher Grundlagen, die Beschreibung des persönlichen Erfahrungshintergrunds der Untersuchungsteilnehmer, die Erhebung der persönlichen Konstrukte sowie die abschließende Kommentierung der erhobenen Konstrukte und der verglichenen Integrationsvorhaben. Die Erhebung der persönli-

⁵³³ Diese Listen beruhen auf Erhebungen des Marktforschungs- und Beratungsunternehmens Lünendonk aus dem Jahr 2004 (vgl. Lünendonk (Hrsg.) /IT-Beratung und Systemintegration/; Lünendonk (Hrsg.) /IT-Services/; Lünendonk (Hrsg.) /Business Innovation and Transformation/).

chen Konstrukte und ihre Kommentierung wurden im Einverständnis mit den Auskunftspersonen auf Tonband aufgezeichnet.

Ungefähr ein Drittel der Interviews wurde in den Räumen des Lehrstuhls für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln durchgeführt. Die anderen Interviews fanden vor Ort in den jeweiligen Unternehmen statt.

4.4.3.1 Einführung und Vereinbarung zur Gesprächsaufzeichnung

Als Einführung in das Gespräch wurden die Zielsetzung der Untersuchung und der geplante Ablauf der Interviews erläutert. Es wurde betont, dass die Untersuchungsteilnehmer jederzeit Fragen oder Anmerkungen äußern und sich Zeit zum Nachdenken nehmen könnten.

Darüber hinaus wurde nochmals das Einverständnis der Untersuchungsteilnehmer eingeholt, die Interviewabschnitte der Erhebung der persönlichen Konstrukte und ihrer Kommentierung auf Tonband aufzuzeichnen. Die Vertraulichkeit und zweckgebundene Verwendung der Aufzeichnungen wurde zugesichert und mit den Untersuchungsteilnehmern vereinbart, ihnen die vollständigen Abschriften der Aufzeichnungen zuzusenden und vor der weiteren Auswertung ihre Zustimmung zu erwarten.

Ferner wurde ein kleines Aufzeichnungsgerät verwendet, das offen sichtbar platziert wurde. Sowohl der Beginn der Aufzeichnung als auch deren Beendigung erfolgte in Abstimmung mit den Untersuchungsteilnehmern. Dabei wurde betont, dass die Aufzeichnung jederzeit auf ihren Wunsch hin unterbrochen werden konnte.⁵³⁴

4.4.3.2 Schaffung begrifflicher Grundlagen

Im nächsten Abschnitt der Interviews sollte ein gemeinsames Verständnis zentraler Begriffe geschaffen werden und eine Verständigung auf den weiteren Gesprächsgegenstand erfolgen. Diesbezüglich erschienen insbesondere die Begriffe Anwendungsintegration, Integrationsvorhaben und Entwicklungsaufwand von Bedeutung, zu denen die in

⁵³⁴ Die beschriebene Vorgehensweise zur akustischen Aufzeichnung der Gespräche greift die Empfehlungen verschiedener Wissenschaftler auf, die von Belanger durch einen Aufruf im Internetportal www.isworld.org gesammelt wurden (vgl. Belanger /Taping Research Interviews/).

dieser Arbeit erläuterten Definitionen und Abgrenzungen als einheitlicher Bezugspunkt der Untersuchungsteilnehmer vereinbart wurde.

In der Erprobung der Untersuchungsmethode hatte sich herausgestellt, dass die Mitarbeiter des Lehrstuhls bei den Vergleichen der Elemente unterschiedliche Betrachtungsperspektiven anwandten, beispielsweise inwiefern nur die fertig gestellten Diplomarbeitsdokumente, die zugrunde liegenden Aufgabenstellungen, die Charakteristika der betreuten Studenten oder weitere Rahmenbedingungen im Umfeld der Betreuungsarbeit in die Betrachtung einbezogen werden sollten. Dabei konnte nicht generell von bewussten Einschränkungen ausgegangen werden, in dem Sinne, dass den ausgegrenzten Einflussfaktoren auf den Betreuungsaufwand eine geringere Bedeutung beigemessen wurde.

In der Untersuchung sollten die Auskunftspersonen daher, in Anlehnung an die von Moynihan beschriebene Vorgehensweise, zu einer möglichst breiten Betrachtungsperspektive angeregt werden.⁵³⁵ Dazu wurde stets im gleichen Wortlaut darauf hingewiesen, dass sämtliche Aspekte, welche den abgegrenzten Entwicklungsaufwand eines Integrationsvorhabens beeinflussen, Gegenstand des weiteren Gesprächs sein sollten: „Diese können mit dem zugrunde liegenden Integrationsproblem, der entworfenen Integrationslösung, den eingesetzten Technologien und beteiligten Personen sowie sonstigen beeinflussbaren oder nicht beeinflussbaren Bedingungen zusammenhängen.“⁵³⁶

Abschließend wurde betont, dass durch die Untersuchung die subjektiven Erfahrungen der Teilnehmer erfasst werden sollten und dass eine Beurteilung ihrer Auskünfte hinsichtlich deren Relevanz oder Richtigkeit weder angestrebt noch möglich sei. Insbesondere wurden die Untersuchungsteilnehmer gebeten, auch solche Aspekte anzusprechen, die ihnen womöglich als trivial erschienen. Auf diese Weise sollte eine möglichst offene Auseinandersetzung der Teilnehmer mit den Integrationsvorhaben erreicht werden.

⁵³⁵ „The researcher then said: ‚Projects can differ from one another in terms of situational factors that developers must take into account when planning and running them. These factors could relate to the client, the deliverable, the resources available, or whatever’.“ (Moynihan /Project Risk/ 360).

⁵³⁶ Dieses Vorgehen greift den Hinweis von Rugg u. a. auf, dass unterschiedliche oder wechselnde Betrachtungsperspektiven der Auskunftspersonen zu vermeintlich inkonsistenten Untersuchungsergebnissen führen können, weswegen auf eine geeignete und einheitliche Frageformulierung geachtet werden sollte (vgl. Rugg u. a. /Laddering/ 220).

4.4.3.3 Erfahrungshintergrund der Untersuchungsteilnehmer

Im folgenden Gesprächsabschnitt wurden einige Fragen zum persönlichen Erfahrungshintergrund der Untersuchungsteilnehmer gestellt. Anhand dieser Informationen soll ein besseres Verständnis davon gewonnen werden, auf welchen Erfahrungen ihre persönlichen Konstrukte basieren.

Es wurde gefragt, seit wie vielen Jahren die Teilnehmer bereits Berufserfahrung im Bereich der Softwareentwicklung haben, an wie vielen Integrationsvorhaben sie bereits teilgenommen haben und für wie viele dieser Projekte sie an der Planung oder Aufwandschätzung mitgewirkt haben.

Ferner wurde eine zusammenfassende Beurteilung verschiedener Merkmale der durchgeführten Integrationsvorhaben erbeten. Diesbezüglich wurde erfragt, in welcher geschätzten Bandbreite diese Projekte hinsichtlich der Anzahl der Mitarbeiter oder der Dauer lagen, in welchen Branchen die Unternehmen tätig sind, deren Anwendungssysteme im Rahmen der Projekte integriert wurden und inwiefern in den Projekten die gleichen Integrationsprodukte oder Technologien eingesetzt wurden (beispielsweise aufgrund exklusiver Partnerschaften mit Herstellern für Integrationsprodukte).

4.4.3.4 Erhebung der persönlichen Konstrukte

Im nächsten Abschnitt der Interviews wurden die persönlichen Konstrukte der Untersuchungsteilnehmer erhoben. Dies geschah anhand von drei Schritten: zunächst wurde die Untersuchungsmethode anhand eines Beispiels erläutert, anschließend wurden mehrere Integrationsvorhaben als Elemente ausgewählt, schließlich wurden die Vergleiche der Integrationsvorhaben gemäß der spezifizierten Vorgehensweise durchgeführt.

▪ **Beispielhafte Erläuterung der Untersuchungsmethode**

Zunächst wurden die Untersuchungsmethode und die zugrunde liegende Theorie der persönlichen Konstrukte anhand eines kurzen Beispiels erläutert. Auf diese Weise sollten die Teilnehmer einen Eindruck davon erhalten, wie sich der weitere Ablauf der Interviews gestalten würde und welche Ergebnisse dabei erarbeitet werden sollten.

Um die späteren Auskünfte der Untersuchungsteilnehmer nicht zu beeinflussen, wurde zu diesem Zweck ein Beispiel aus einem anderen Themenbereich gewählt: als Elemente

wurden zwei unterschiedliche Automodelle gegenübergestellt und anhand einer bereits ausgefüllten Karte die Erhebung eines Konstrukts zur Beurteilung des Kofferraumvolumens nachgestellt.

No. F1	Vergleich von: <i>Kleinwagen & Kleintransporter</i>	
<i>Großes Kofferraumvolumen</i>		<i>Kein Kofferraum</i>
<div style="display: flex; justify-content: space-around; align-items: center;"><div style="text-align: center;"><input checked="" type="checkbox"/></div><div style="text-align: center;">Ziehe ich bei der Auswahl eines Autos vor</div><div style="text-align: center;"><input type="checkbox"/></div></div>		

Abbildung 4-1: Darstellung der Konstruktkarte zur Erläuterung der Untersuchungsmethode gegenüber den Auskunftspersonen

▪ Auswahl der Integrationsvorhaben

Als nächstes wurden die Untersuchungsteilnehmer gebeten, verschiedene Integrationsvorhaben zu bestimmen, die im weiteren Gespräch betrachtet werden sollten. Dazu wurden ihnen die spezifizierten Anforderungen an die Auswahl der Integrationsvorhaben erläutert, an denen sie sich orientieren sollten.

Die Teilnehmer notierten darauf hin zu jedem ausgewählten Integrationsvorhaben eine für sie verständliche Bezeichnung auf eine separate Elementkarte. Um die Durchführung der Vergleiche zu erleichtern, verfügte jede Elementkarte über eine eindeutige Nummer. Darüber hinaus dokumentierten die Untersuchungsteilnehmer, in welchem

Zeitraum das jeweilige Projekt durchgeführt wurde und in welcher Rolle bzw. mit welcher Zuständigkeit sie daran mitgewirkt hatten.

No. P1		Rang bzgl. Aufwand:
Bezeichnung:		
Zeitraum:		
Ihre Rolle / Zuständigkeit im Projekt:		

Abbildung 4-2: Darstellung einer unbeschrifteten Elementkarte

Nachdem die Beschriftung der Elementkarten für die ausgewählten Integrationsvorhaben abgeschlossen war, wurde mit der Tonbandaufzeichnung des Gesprächs begonnen.

Die Untersuchungsteilnehmer wurden gebeten, jedes der ausgewählten Integrationsvorhaben in wenigen Sätzen zu beschreiben. Auf diese Weise erhielt der Interviewer eine grobe Vorstellung von den betrachteten Projekten und die Teilnehmer konnten sich die grundlegenden Charakteristika der Projekte in Erinnerung rufen.

Anschließend sollten die Integrationsvorhaben durch die Untersuchungsteilnehmer hinsichtlich des mit ihnen verbundenen Entwicklungsaufwands in eine Reihenfolge gebracht werden. Die jeweilige Position der Integrationsvorhaben in dieser Reihenfolge wurde auf den Elementkarten notiert („Rang bzgl. Aufwand“). Auf diese Weise wurde eine weitere Orientierungshilfe für den anschließenden Vergleich der Projekte geschaf-

fen, denn die Teilnehmer konnten sich nunmehr stets die unterschiedliche Beurteilung des Entwicklungsaufwands vor Augen führen.

▪ **Vergleiche der Integrationsvorhaben**

Für die Erhebung der persönlichen Konstrukte wurden darauf hin systematisch Paarungen zwischen zwei der Integrationsvorhaben ausgewählt und die betreffenden Elementkarten vor die Untersuchungsteilnehmer gelegt.

Wiederum sollte durch eine gezielte und stets gleiche Formulierung der Fragestellung sichergestellt werden, dass die Teilnehmer eine angemessene und einheitliche Betrachtungsperspektive beim Vergleich der Integrationsvorhaben einnahmen.⁵³⁷ Der Vergleich der Integrationsvorhaben wurde in diesem Sinne durch die folgende Fragestellung eingeleitet, die auch im weiteren Verlauf der Erhebung der Konstrukte immer wieder aufgegriffen wurde: ‚Projekte können sich im Hinblick auf verschiedene Faktoren unterscheiden, die Einfluss auf den benötigten Entwicklungsaufwand für die Realisierung und Inbetriebnahme einer entworfenen Integrationslösung haben. Diese können beispielsweise mit dem zugrunde liegenden Integrationsproblem, der entworfenen Integrationslösung, den eingesetzten Technologien und beteiligten Personen sowie sonstigen beeinflussbaren oder nicht beeinflussbaren Bedingungen zusammenhängen. Bitte nennen Sie einen Unterschied zwischen diesen beiden Projekten, der Ihrer Meinung nach zu einem unterschiedlichen Entwicklungsaufwand für die Realisierung und Inbetriebnahme der entworfenen Integrationslösung geführt hat.‘

Sofern die Untersuchungsteilnehmer einen relevanten Unterschied zwischen den verglichenen Integrationsvorhaben identifizieren konnten, wurden entsprechend der spezifizierten Vorgehensweise die beiden polaren Ausprägungen des dahinter stehenden Konstrukts bestimmt. Diese wurden im Wortlaut der Teilnehmer auf Konstruktkarten notiert und zudem vermerkt, welcher Konstruktpol aus ihrer Sicht mit einem höheren Entwicklungsaufwand verbunden ist.

⁵³⁷ Wie bereits erläutert wurde, greift dieses Vorgehen den Hinweis von Rugg u. a. auf, dass unterschiedliche oder wechselnde Betrachtungsperspektiven der Auskunftspersonen zu vermeintlich inkonsistenten Untersuchungsergebnissen führen können, weswegen auf eine geeignete und einheitliche Frageformulierung geachtet werden sollte (vgl. Rugg u. a. /Laddering/ 220).

Zudem dokumentierte der Interviewer auf den Konstruktkarten die eindeutigen Nummern der Elementkarten der jeweils verglichenen Integrationsvorhaben. Auf diese Weise konnte bei der späteren Besprechung eines Konstruktes der ursprüngliche Gesprächskontext rekonstruiert werden. Darüber hinaus waren die Konstruktkarten ihrerseits mit eindeutigen Nummern versehen, um Bezugnahmen im weiteren Verlauf der Interviews zu erleichtern.

Die Bestimmung der Konstruktpole und die Festlegung auf die zu notierenden Formulierungen erfolgten teilweise eigenständig durch die Untersuchungsteilnehmer. In vielen Fällen entwickelten die Teilnehmer jedoch erst im Dialog mit dem Interviewer ein präzises Bewusstsein für ihr implizites Verständnis von den maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand. Dabei wurde große Sorge getragen, die Auskünfte der Interviewpartner nicht zu verfälschen.⁵³⁸

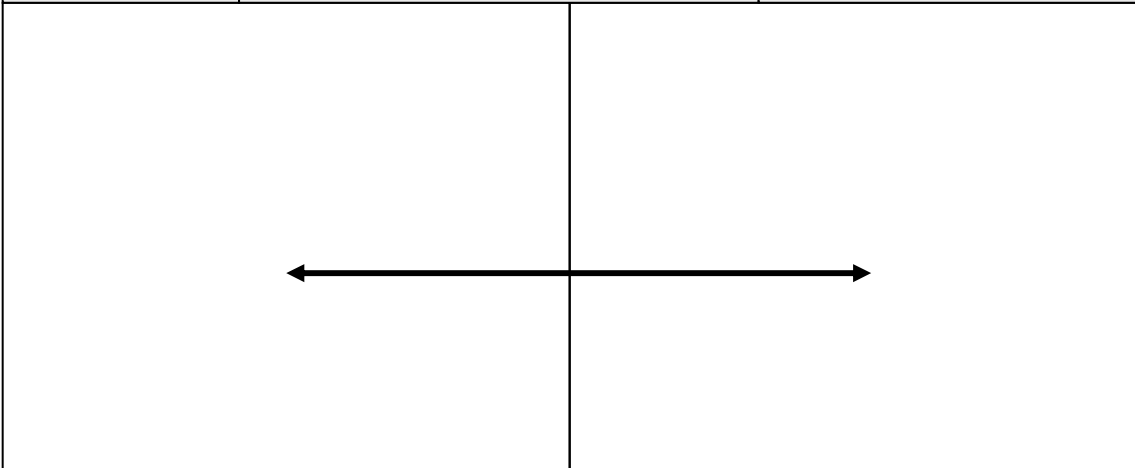
No. F1	Vergleich der Projekte:	Rang bzgl. Aufwand:
		
<input type="checkbox"/>	Ist mit größerem Aufwand verbunden.	<input type="checkbox"/>
<input type="checkbox"/> Dieses Merkmal ist charakteristisch für Integrationsvorhaben.		

Abbildung 4-3: Darstellung einer unbeschrifteten Konstruktkarte

⁵³⁸ Vgl. diesbezüglich die Ausführungen zur Vorbereitung der Untersuchung in Kapitel 3.4.1.

Anschließend wurden die Untersuchungsteilnehmer in gleicher Weise gebeten, einen weiteren relevanten Unterschied zu bestimmen. Sofern sie keinen weiteren relevanten Unterschied zwischen den verglichenen Integrationsvorhaben identifizieren konnten oder lediglich solche Merkmale der Projekte auffielen, deren entsprechende Konstrukte bereits erfasst worden waren, wurde eine neue Paarung unter den Integrationsvorhaben gebildet und der Vergleich von neuem begonnen.

Es gab keinerlei Vorgaben, wie viele Unterschiede zwischen zwei Integrationsvorhaben identifiziert bzw. wie viele Konstrukte im Rahmen eines Interviews notiert werden sollten. Auch wurden keine Angaben dazu gemacht, wie viele Konstrukte in anderen Interviews erhoben worden waren oder auf welche Aspekte von Integrationsvorhaben sich diese beziehen. Die Teilnehmer führten die Vergleiche somit in eigenem Ermessen durch. Dabei stand es ihnen stets frei, die bereits beschrifteten Konstruktkarten einzusehen.

Sobald alle möglichen Paarungen unter den Integrationsvorhaben durch die Auskunftspersonen verglichen worden waren, fragte der Interviewer, ob über die auf den Konstruktkarten notierten Merkmale hinaus noch weitere Charakteristika der betrachteten Integrationsvorhaben als relevant für ihren Entwicklungsaufwand angesehen werden. In wenigen Fällen wurden daraufhin weitere Konstrukte erfasst.⁵³⁹

Insgesamt dauerten der Vergleich der Integrationsvorhaben und die Erhebung der Konstrukte jeweils zwischen einer und anderthalb Stunden.

4.4.3.5 Kommentierung der erhobenen Konstrukte

Zu Beginn des nächsten Interviewabschnitts wurden die Konstruktkarten zunächst in eine zufällige Reihenfolge gebracht, um zu vermeiden, dass die Reihenfolge ihrer Nennung nicht ihre weitere Kommentierung beeinflussen würde.

⁵³⁹ Diese Vorgehensweise entspricht dem von Moynihan beschriebenen Vorgehen (vgl. Moynihan /Project Risk/ 360).

Darauf hin wurden die Untersuchungsteilnehmer gebeten, die Konstruktkarten hinsichtlich der Bedeutung der auf ihnen beschriebenen Merkmale für den Entwicklungsaufwand in drei Gruppen einzuteilen: Merkmale mit einer hohen Bedeutung für den Entwicklungsaufwand, Merkmale mit mittlerer Bedeutung für den Entwicklungsaufwand und Merkmale mit niedriger Bedeutung für den Entwicklungsaufwand. Dabei wurde darauf hingewiesen, dass der Einfluss der beschriebenen Merkmale auf den Entwicklungsaufwand isoliert voneinander und unter Bezug auf den markierten aufwandstreibenden Konstruktpol eingeschätzt werden sollte. Zudem wurde betont, dass die Verteilung der Konstruktkarten gänzlich nach dem Ermessen der Untersuchungsteilnehmer erfolgen sollte.⁵⁴⁰ Die jeweilige Einordnung wurde auf den Konstruktkarten notiert („Rang bzgl. Aufwand“).

Des Weiteren wurden die Untersuchungsteilnehmer gebeten, diejenigen Konstruktkarten zu benennen, deren beschriebene Merkmale sie als charakteristisch für Integrationsvorhaben ansehen. Auf diese Weise soll ein besseres Verständnis der Besonderheiten der Integrationsvorhaben gegenüber anderen Arten von Softwareentwicklungsvorhaben ermöglicht werden. Auch diese Auskunft wurde auf den entsprechenden Konstruktkarten vermerkt.

Sofern noch Gesprächszeit mit den Teilnehmern übrig blieb, wurden im Weiteren diejenigen Konstruktkarten ausgewählt, denen sie eine hohe Bedeutung für den Entwicklungsaufwand zugemessen hatten. Die Teilnehmer wurden darauf hin jeweils gebeten zu erläutern, warum der entsprechend markierte Konstruktpol gemäß ihrer Erfahrung mit einem höheren Entwicklungsaufwand verbunden ist.

Nachdem die Kommentierung der Konstrukte abgeschlossen war, wurde die Tonbandaufzeichnung des Gesprächs beendet.

4.4.3.6 Kommentierung der betrachteten Integrationsvorhaben

Es wurde bereits mehrfach darauf hingewiesen, dass in der angewandten Untersuchungsmethode die verglichenen Elemente determinieren, welche Konstrukte der Aus-

⁵⁴⁰ Beispielsweise wurde es den Untersuchungsteilnehmern freigestellt, einzelnen Kategorien keine Konstruktkarte zuzuordnen.

kunftspersonen erhoben werden. Nicht nur der individuelle Erfahrungshintergrund der Auskunftspersonen, auch die Arten der durch sie verglichenen Integrationsvorhaben sollten daher erfasst werden, um die Untersuchungsergebnisse interpretieren zu können.⁵⁴¹

Im letzten Abschnitt des Interviews wurden somit grundlegende Merkmale der verglichenen Integrationsvorhaben erhoben. Dazu wurden die Untersuchungsteilnehmer gebeten, auf der Rückseite der Elementkarten einige, überwiegend geschlossene Fragen zu den einzelnen Integrationsvorhaben schriftlich zu beantworten. Die Auswahl der erfragten Merkmale der Integrationsvorhaben orientiert sich an den vorgestellten Ansätzen zur Abgrenzung verschiedener Kategorien von Integrationslösungen.⁵⁴²

Um die Erhebung der Konstrukte nicht zu beeinflussen wurden ihnen diese Fragen erst jetzt, am Ende der Interviews, offen gelegt.

Zunächst wurde erfasst, inwiefern mit dem jeweiligen Integrationsvorhaben eine informationsorientierte, eine funktionsorientierte bzw. serviceorientierte und eine geschäftsprozessorientierte Anwendungsintegration angestrebt wurde. Zu jeder dieser Integrationsperspektiven wurde ein Item als Aussage formuliert, zu denen die Untersuchungsteilnehmer ihre Zustimmung auf fünfstufigen bipolaren Rating-Skalen mit den Extremitäten „trifft nicht zu“ und „trifft voll zu“ markieren sollten.⁵⁴³

Darüber hinaus wurde die jeweilige Integrationsreichweite in einem geschlossenen Item charakterisiert. Dazu sollten die Untersuchungsteilnehmer eine von verschiedenen Ausprägungen der inner- und zwischenbetrieblichen Anwendungsintegration auswählen, die als Stufen einer zunehmenden Integrationsreichweite verstanden werden können.

In gleicher Weise wurde erfragt, ob Anwendungssysteme von Organisationseinheiten innerhalb eines Landes oder über Landesgrenzen hinweg integriert wurden.

⁵⁴¹ In dieser Hinsicht erscheint es als Mangel der von Moynihan beschriebenen Studie, dass die betrachteten Softwareentwicklungsvorhaben nicht näher charakterisiert werden (vgl. Moynihan/ Project Risk/ 360).

⁵⁴² Diesbezüglich kann wiederum angemerkt werden, dass nicht bekannt ist, welche Merkmale der Projekte im Zusammenhang mit dem erforderlichen Entwicklungsaufwand von Bedeutung und daher für die Interpretation der Untersuchungsergebnisse entscheidend sind.

⁵⁴³ Zur Formulierung und Skalierung von Items vgl. beispielsweise Müller /Prüf- und Testprozesse/ 119-123.

Ferner sollten die Arten der integrierten Anwendungssysteme benannt werden. Dazu sollten die Untersuchungsteilnehmer angeben, ob ERP-Systeme, Altsysteme, sonstige standardisierten Anwendungssysteme, sonstige individuell entwickelte Anwendungssysteme oder internetbasierte Anwendungen verknüpft wurden (Mehrfachnennung war möglich). Ergänzend sollten die Teilnehmer einschätzen, welche Anzahl von Anwendungssystemen in den Integrationsvorhaben verknüpft wurde.

Schließlich sollte durch ein letztes Item erfasst werden, inwiefern bei der Realisierung des angestrebten Integrationszustands auf bereits bestehenden Integrationsmaßnahmen aufgebaut werden konnte. Dazu sollten die Untersuchungsteilnehmer eine von drei verschiedenen Ausprägungen auswählen, die als Stufen einer zunehmenden Wiederverwendung früherer Integrationsmaßnahmen verstanden werden können.




Das Projekt zielte insbesondere darauf Informationen zwischen den betroffenen Anwendungssystemen auszutauschen. <div style="text-align: right;"> Trifft nicht zu  Trifft voll zu </div>	
... einzelne Funktionen der betroffenen Anwendungssysteme verfügbar zu machen (z. B. als Services) und sie flexibel zur neuartigen Unterstützung von Anwendungsgebieten kombinieren zu können. <div style="text-align: right;"> Trifft nicht zu  Trifft voll zu </div>	
... mehrstufige automatisierte Interaktionsfolgen zwischen den betroffenen Anwendungssystemen zur Abwicklung von Geschäftsprozessen zu realisieren. <div style="text-align: right;"> Trifft nicht zu  Trifft voll zu </div>	
Das Projekt betraf Anwendungssysteme ... <input type="checkbox"/> innerhalb einer Abteilung eines Unternehmens. <input type="checkbox"/> innerhalb mehrerer Abteilungen eines Unternehmens. <input type="checkbox"/> innerhalb mehrerer Teilgesellschaften eines Konzerns. <input type="checkbox"/> innerhalb mehrerer unabhängiger Unternehmen	Das Projekt betraf Organisationen ... <input type="checkbox"/> innerhalb eines Landes. <input type="checkbox"/> innerhalb mehrerer Länder.
Arten der betroffenen Anwendungssysteme (Mehrfachnennung möglich) <input type="checkbox"/> ERP-Systeme <input type="checkbox"/> Alt-Systeme <input type="checkbox"/> Sonstige Standard-Anwendungssysteme <input type="checkbox"/> Sonstige Individual-Anwendungssysteme <input type="checkbox"/> Internetbasierte Anwendungen (z. B. eStores, eServices)	(ungefähre) Anzahl der betroffenen Anwendungssysteme:
Im Rahmen des Projekts ... <input type="checkbox"/> wurde eine vollständig neue Integrationsarchitektur implementiert. <input type="checkbox"/> wurde eine bestehende Integrationsarchitektur modifiziert oder erweitert. <input type="checkbox"/> wurden weitere Anwendungssysteme an eine bestehende Integrationsinfrastruktur angebunden oder neue Interaktionen zwischen bereits angehenden Anwendungssystemen realisiert	

Abbildung 4-4: Darstellung der Rückseite einer unbeschrifteten Elementkarte

4.4.3.7 Rückmeldung der Untersuchungsteilnehmer

Die Rückmeldung der Untersuchungsteilnehmer zu der Durchführung der Interviews und der angewandten Untersuchungsmethode war durchweg positiv. Die Methode wur-

de als anregend, neuartig und überzeugend beurteilt. Negative Erfahrungen, wie sie beispielsweise von Brown beschrieben werden, können nicht bestätigt werden.⁵⁴⁴

Die Untersuchungsteilnehmer standen den paarweisen Vergleichen der Integrationsvorhaben und der Verwendung der Element- und Konstruktkarten offen gegenüber. Viele von ihnen freuten sich über die Gelegenheit, vergangene Projekte zu reflektieren. Nicht selten waren sie dabei erstaunt, welche Einflussfaktoren auf den Entwicklungsaufwand ihnen erst im Zuge der Gespräche bewusst wurden. Diese Rückmeldung der Teilnehmer kann als Hinweis angesehen werden, dass es tatsächlich gelungen ist, auch Teile ihres impliziten Wissens zu erschließen.

Eine Beeinflussung des Antwortverhaltens durch die Aufzeichnung der Gespräche, welche die Aussagekraft der erhobenen Informationen in Frage stellen könnte, wurde nicht festgestellt. Angesichts der getroffenen Vereinbarungen schienen sich die Untersuchungsteilnehmer unbefangen mit den betrachteten Integrationsvorhaben auseinanderzusetzen.

4.4.4 Transkription der Gesprächsaufzeichnungen

Im Zuge der Nachbereitung der Interviews wurden die Tonbandaufzeichnungen in eine schriftliche Form überführt. Die auf diese Weise erstellten Transkripte sollten eine angemessene Interpretation der auf den Konstruktkarten notierten Formulierungen ermöglichen.⁵⁴⁵

Angesichts dieser Zielsetzung galt es in den Transkripten vor allem solche Gesprächsinhalte zu dokumentieren, welche Aussagen zu Einflussfaktoren und ihren Wirkungszusammenhängen beinhalten. Auf die Erfassung der lautlichen Gestaltung dieser Aussagen (z. B. Tonhöhe und Lautstärke) sowie des redegleitenden nichtsprachlichen Verhaltens (z. B. Lachen oder Räuspern) wurde verzichtet und die Methode der Paraphrasierung angewandt. Auf diese Weise konnte auch von einem komplexen Katalog von Notationszeichen abgesehen werden. Darüber hinaus wurde als Transkriptionsformat die Zeilenschreibweise verwendet, d. h. aufeinander folgende Gesprächsbeiträge wurden jeweils mit einer neuen Transkriptzeile begonnen. Insgesamt sollten damit sowohl

⁵⁴⁴ Vgl. Brown /Qualitative Survey Research/ 302.

⁵⁴⁵ Darüber hinaus kann durch eine zukünftige qualitative Inhaltsanalyse der Transkripte versucht werden, weitere Konstrukte und die zwischen ihnen bestehenden Beziehungen zu extrahieren.

das Verfassen als auch das Lesen der Transkripte deutlich erleichtert und eine zweckdienliche Reduktion der aufgezeichneten Informationen erreicht werden.⁵⁴⁶

Aufbauend auf diesen Zielsetzungen, wurde beim Erstellen der Transkripte wie folgt vorgegangen:

Zunächst wurde ein Leitfaden für die Erstellung der Transkripte aufgestellt und die zuständigen Personen eingewiesen, um eine einheitliche Qualität der Abschriften zu erreichen.

Es erfolgte keine Dokumentation von Versprechern, Gesprächspausen, Lachen und sachfremden Exkursen. Es wurde auch keine getreue Wiedergabe von Dialekten angestrebt sondern die Regeln der Standardorthographie angewandt.

Im Interesse der Anonymität wurden Namen von Personen, Unternehmen oder spezifischen Produkten nicht dokumentiert. Sollte aus dem Gesprächskontext ein Rückschluss auf Personen oder Unternehmen möglich erscheinen, wurde die entsprechende Passage ebenfalls nicht oder nur in abstrahierter Form dokumentiert.

Abgesehen von den vorangehenden Maßnahmen wurde große Sorgfalt darauf verwendet, die sachbezogenen Gesprächsinhalte im exakten Wortlaut der Gesprächspartner zu dokumentieren. Zu diesem Zweck wurde jedes Transkript durch eine zweite Person (den Interviewer) gegen die betroffene Tonbandaufzeichnung geprüft und nötigenfalls ergänzt oder angepasst.

Da im Rahmen der Gespräche durch den Einsatz der Element- und Konstruktkarten oftmals Gestiken zur Erläuterung von Sachverhalten verwendet wurden, wurden in einzelnen Fällen Betonungen durch die Unterstreichung von Wörtern gekennzeichnet, um das Verständnis der Aussagen zu unterstützen.

Nach der Fertigstellung eines Transkriptes wurde es stets dem jeweiligen Untersuchungsteilnehmer zugesandt, damit dieser die exakte Wiedergabe seiner Ansichten und Erfahrungen sowie die vorgenommene Anonymisierung überprüfen konnte. In wenigen Fällen wurden auf Wunsch der Teilnehmer kleinere Änderungen in den Transkripten vorgenommen. Die zugrunde liegenden Tonbandaufzeichnungen der Interviews wurden anschließend vereinbarungsgemäß gelöscht.

⁵⁴⁶ Zur Transkription von Gesprächsaufzeichnungen vgl. Flick /Qualitative Forschung/ 438-445.

Wie erwartet, erwies sich die Abschrift der Gespräche als äußerst zeitintensiv. Die Tonbandaufzeichnungen umfassten jeweils eine bis anderthalb Stunden Gesprächszeit. Das Erstellen und Überprüfen der einzelnen Abschriften erforderte daraufhin jeweils ca. zehn bis vierzehn Arbeitsstunden der beteiligten Personen.

5 Ergebnisse der Untersuchung

Der Beschreibung der Untersuchungssubjekte und -objekte bzw. der Auskunftspersonen und der durch sie verglichenen Integrationsvorhaben wird eine große Bedeutung für die Interpretation der Untersuchungsergebnisse beigemessen.⁵⁴⁷ Sie erfolgt in den beiden folgenden Kapiteln. Anschließend werden die erhobenen Konstrukte systematisch dargestellt und dem im 3. Kapitel beschriebenen bisherigen Stand der Erkenntnis gegenübergestellt.

5.1 Charakterisierung der Untersuchungsteilnehmer

Insgesamt nahmen 22 Personen an der Untersuchung teil.⁵⁴⁸ Im Weiteren werden die erhobenen Informationen zu ihrem Erfahrungshintergrund dargestellt. Anhand dieser Angaben wird deutlich, dass die spezifizierten Anforderungen an den Erfahrungshintergrund der Auskunftspersonen vollständig erfüllt wurden.

▪ Größe der Unternehmen

Die 22 Untersuchungsteilnehmer stammen aus 10 verschiedenen Unternehmen aus der IT-Branche. Diese Unternehmen erbringen überwiegend Dienstleistungen für externe Auftraggeber, lediglich zwei von ihnen führen insbesondere konzerninterne Entwicklungsvorhaben durch. Allein 7 der Unternehmen können gemäß dem Marktforschungs- und Beratungsunternehmen Lünendonk zu den Top Unternehmen für IT-Beratung- und Systemintegration, IT-Services oder Business Innovation/Transformation in Deutschland gezählt werden.⁵⁴⁹

⁵⁴⁷ Kitchenham u. a. /Empirical Research/ 725.

⁵⁴⁸ Sämtliche Teilnehmer sind männlich.

⁵⁴⁹ Diese Angaben basieren auf den sog. Lünendonk-Listen des Marktforschungs- und Beratungsunternehmens aus dem Jahr 2005 (vgl. Lünendonk (Hrsg.) /IT-Beratung und Systemintegration/; Lünendonk (Hrsg.) /IT-Services/; Lünendonk (Hrsg.) /Business Innovation and Transformation/). Demgegenüber stammen die Untersuchungsteilnehmer der bereits erwähnten Studie von Moynihan ausschließlich aus kleinen Unternehmen mit bis zu 10 Mitarbeitern (vgl. Moynihan /Project Risk/ 360).

▪ Berufserfahrung in der Softwareentwicklung

Alle Untersuchungsteilnehmer besitzen eine mehrjährige Berufserfahrung in der Softwareentwicklung. Die Hälfte von ihnen gab 6 bis 10 Jahre an Erfahrung an (50 %), vier der Untersuchungsteilnehmer arbeiten seit 11 bis 15 Jahren (18 %), drei von ihnen seit 16 bis 20 Jahren (14 %) und vier von ihnen seit 21 bis 25 Jahren (18 %) in der Softwareentwicklung. Im Durchschnitt besitzen die Untersuchungsteilnehmer damit 13 Jahre Berufserfahrung in der Softwareentwicklung.

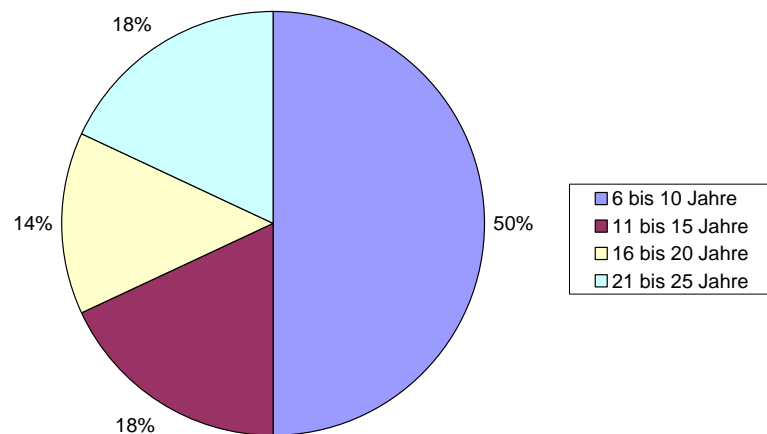


Abbildung 5-1: Berufserfahrung der Untersuchungsteilnehmer in Jahren

▪ Projekterfahrung in der Anwendungsintegration

Alle Untersuchungsteilnehmer haben bereits an mehreren Integrationsvorhaben mitgewirkt. Zehn von ihnen gaben an, dass sie an bis zu 5 Integrationsvorhaben mitgearbeitet haben (45 %), vier von ihnen nannten 6 bis 10 Projekte (18 %) und fünf von ihnen 11 bis 15 Projekte (23 %). Zwei der Teilnehmer gaben ferner an, dass sie an 21 bis 25 Integrationsvorhaben mitgearbeitet haben, und ein Teilnehmer verwies auf mehr als 25 Projekte (5 %).⁵⁵⁰ Zusammenfassend können die Untersuchungsteilnehmer damit als sehr erfahren angesehen werden.⁵⁵¹

⁵⁵⁰ Die Interpretation einzelner Werte wird jedoch dadurch erschwert, dass eine einheitliche Abgrenzung der Projekte nicht sichergestellt werden konnte. Die genannten Projekte können zudem von sehr unterschiedlicher Größe sein. Schließlich gaben verschiedene Untersuchungsteilnehmer an, dass sie aufgrund ihrer Zuständigkeiten in ihrem Unternehmen in weitere Projekte Einblicke erhalten haben, was ihr Verständnis der maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand ebenfalls beeinflusst haben kann.

⁵⁵¹ MacCormack, Verganti und Iansiti erläutern eine Untersuchung zur Gestaltung von Softwareentwicklungsvorhaben, in der die Erfahrung der Mitarbeiter ebenfalls anhand der Anzahl der zuvor durchgeführten Entwicklungsvorhaben gemessen wird. Die höchste abgegrenzte Erfahrungsstufe um-

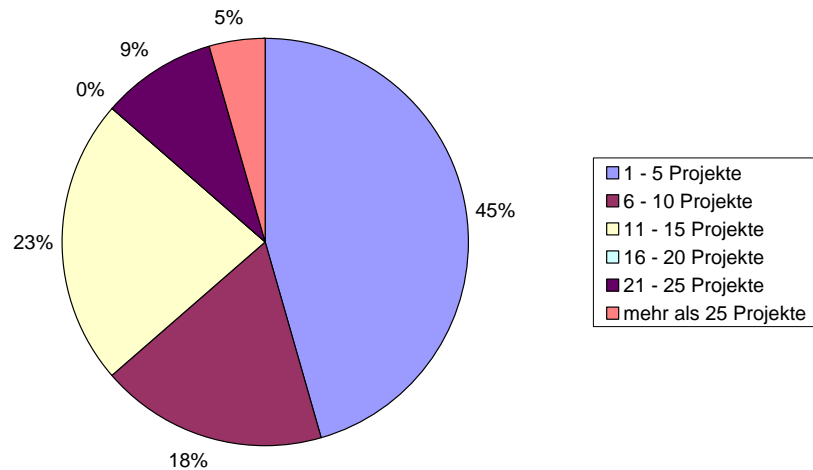


Abbildung 5-2: Anzahl der Integrationsvorhaben, an denen die Untersuchungsteilnehmer mitgewirkt haben

▪ Planungserfahrung in der Anwendungsintegration

Lediglich einer der Untersuchungsteilnehmer gab an, für keines der Integrationsvorhaben an der Planung oder Aufwandschätzung mitgewirkt zu haben (5 %). 13 der Untersuchungsteilnehmer hatten in bis zu 5 Projekten an der Planung oder Aufwandschätzung mitgewirkt (58 %), vier von ihnen in 6 bis 10 Projekten (18 %) und zwei Untersuchungsteilnehmer in 11 bis 15 Projekten (9 %). Des Weiteren basiert die Planungserfahrung eines Untersuchungsteilnehmers auf 16 bis 20 Projekten (5 %) und ein Untersuchungsteilnehmer hat für mehr als 20 Projekte an der Planung oder Aufwandschätzung mitgewirkt (5 %). Die Untersuchungsteilnehmer können damit insgesamt als erfahren in der Planung bzw. in der Aufwandschätzung von Integrationsvorhaben angesehen werden.

fasst dabei Mitarbeiter, die in mehr als zwei vorangehenden unterschiedlichen Entwicklungsvorhaben teilgenommen haben (vgl. MacCormack, Verganti, Iansiti /Flexible Development Process/ 137f., 141f.).

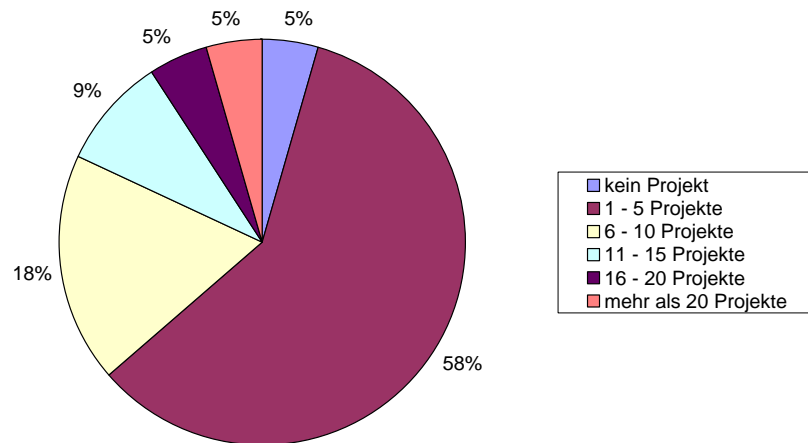


Abbildung 5-3: Anzahl der Integrationsvorhaben, an deren Planung die Untersuchungsteilnehmer mitgewirkt haben

▪ Dauer und Größe der durchgeführten Integrationsvorhaben

Die meisten Untersuchungsteilnehmer konnten Angaben zur Größe der Integrationsvorhaben machen, an denen sie mitgewirkt haben.

Zunächst schätzten sie die Dauer des kürzesten und des längsten Projektes ein. Demnach dauerten die Projekte von drei Tagen (lediglich eine unkomplizierte Schnittstelle musste implementiert werden) bis hin zu vier Jahren. Die größte Überlappung der genannten Intervalle besteht bezüglich Projektdauern von einem halben Jahr bis zu anderthalb Jahren. Es kann daher angenommen werden, dass die von den Untersuchungsteilnehmern durchgeführten Integrationsvorhaben überwiegend eine dazwischen liegende Dauer aufweisen.⁵⁵²

In gleicher Weise schätzten die Untersuchungsteilnehmer die Bandbreite der Anzahl der Mitarbeiter der Integrationsvorhaben ein. Demnach reichten die Integrationsvorhaben von einem Mitarbeiter bis zu 500 Mitarbeitern.⁵⁵³

⁵⁵² Die Interpretation einzelner Werte erscheint wiederum als schwierig. Es konnte nicht sichergestellt werden, ob die Dauer der Projekte in gleicher Weise abgegrenzt wurde (Wurden Warteperioden berücksichtigt, die beispielsweise aus Abhängigkeiten von anderen Projekten anfielen? Wie wurde die Dauer abgegrenzt, wenn das Integrationsvorhaben ein Teilaspekt eines größeren umfassenden Projektes war?).

⁵⁵³ Demgegenüber basieren die Erfahrungen der Teilnehmer der erwähnten Studie von Moynihan lediglich auf Entwicklungsvorhaben mit bis zu 6 Mitarbeitern (vgl. Moynihan /Project Risk/ 360). Es soll

▪ Branchen der Auftraggeber

Die Integrationsvorhaben, an denen die Untersuchungsteilnehmer mitgewirkt haben, wurden in verschiedenen Branchen durchgeführt. Die meisten der Untersuchungsteilnehmer haben Projekte in der Finanz- und Versicherungsbranche durchgeführt, gefolgt von der Telekommunikations- und IT-Branche.

Branche	Anzahl (Anteil) der Untersuchungsteilnehmer, die in dieser Branche mindestens ein Integrationsvorhaben durchgeführt haben.	
Banken/Versicherung	16	(73 %)
Telekommunikation/IT	9	(41 %)
Logistik/Transport	7	(32 %)
Automobilindustrie	5	(23 %)
Vertrieb/Handel	5	(23 %)
Öffentlicher Sektor	5	(23 %)
Dienstleistungen	4	(18 %)
Chemie/Pharma	3	(14 %)
Gesundheitswesen	3	(13 %)
Sonstiges	3	(13 %)

Tabelle 5-1: Branchen, in denen die Auskunftspersonen Integrationsvorhaben durchgeführt haben

▪ Die Variation der eingesetzten Technologien

Die Mehrzahl der Untersuchungsteilnehmer gab an, dass in den durchgeführten Integrationsvorhaben einzelfallbezogen unterschiedliche Technologien eingesetzt wurden. Sechs der Untersuchungsteilnehmer (27 %) berichteten dagegen, dass alle von ihnen durchgeführten Integrationsvorhaben im Wesentlichen auf den gleichen Technologien und insbesondere den gleichen Integrationsprodukten basierten.

darauf hingewiesen werden, dass die Untersuchungsteilnehmer auch bei der Abgrenzung der Anzahl der Projektmitarbeiter in unterschiedlicher Weise vorgegangen sein können (Wurden Mitarbeiter anderer Organisationen einbezogen? Wurden lediglich Mitarbeiter berücksichtigt, die an der Entwicklung und Bereitstellung der Integrationslösung mitwirkten?).

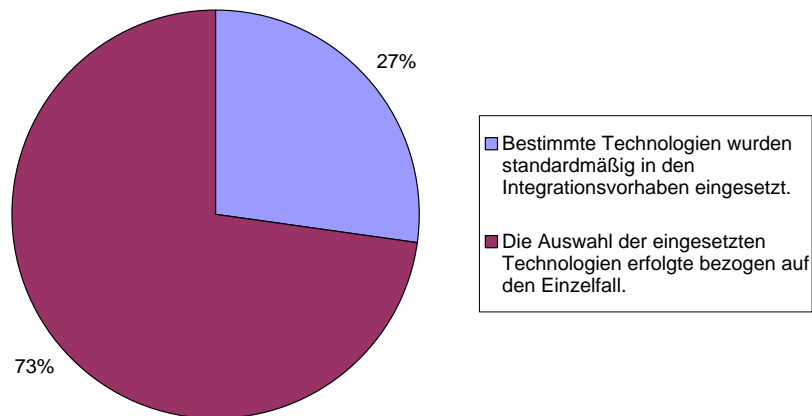


Abbildung 5-4: Variation der eingesetzten Technologien

5.2 Charakterisierung der betrachteten Integrationsvorhaben

Insgesamt wurden im Rahmen der Studie 82 Integrationsvorhaben auf Elementkarten notiert. Im Durchschnitt hat jeder Untersuchungsteilnehmer ca. 4 Integrationsvorhaben miteinander verglichen, so dass die angestrebte Anzahl der verglichenen Elemente insgesamt erreicht wurde.⁵⁵⁴

Im Weiteren werden die erhobenen Informationen zu den Integrationsvorhaben dargestellt. Sie sprechen dafür, dass in der Untersuchung eine große Vielfalt von Integrationsvorhaben betrachtet wurde, so dass die gewonnenen Untersuchungsergebnisse den gesamten Bereich der Anwendungsintegration repräsentieren dürften.

▪ Dauer der betrachteten Integrationsvorhaben

Der größte Teil der betrachteten Integrationsvorhaben wies gemäß den Angaben der Untersuchungsteilnehmer eine Dauer von bis zu 12 Monaten auf (62 %). Des Weiteren gaben die Untersuchungsteilnehmer bei 21 % der Integrationsvorhaben eine Dauer zwischen 12 und 24 Monaten an. Diese Angaben entsprechen damit dem im vorangehenden Kapitel genannten Intervall der typischen Projektdauer. Ferner nannten die Teilnehmer

⁵⁵⁴ Trotz der detaillierten Hinweise zur Auswahl der Integrationsvorhaben konnte eine vollständig einheitliche Abgrenzung der Projekte womöglich nicht sichergestellt werden. Teilweise kann es sich bei den betrachteten Integrationsvorhaben somit um eigenständige Projekte, teilweise um Teilprojekte eines Programms und teilweise um ganze Programme handeln, die zusammenfassend betrachtet wurden.

bei 11 % der Integrationsvorhaben eine Dauer zwischen 25 bis 36 Monaten und bei 6 % der Integrationsvorhaben eine Dauer von mehr als 36 Monaten.

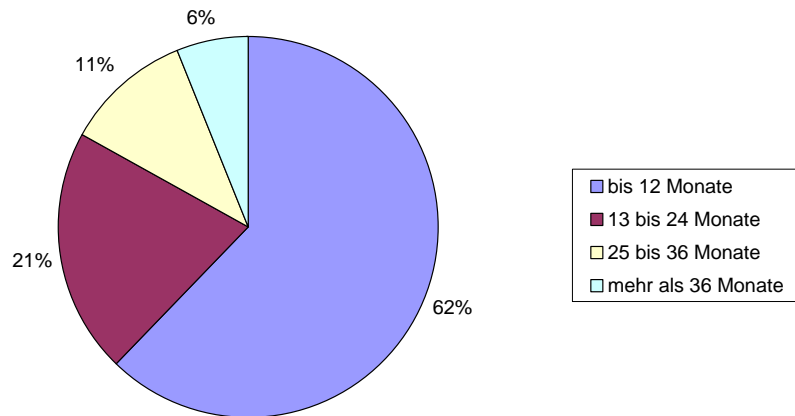


Abbildung 5-5: Dauer der in der Untersuchung betrachteten Integrationsvorhaben

▪ **Abschlussjahr der betrachteten Integrationsvorhaben**

Es wurde bereits erläutert, dass es gemäß der Theorie der persönlichen Konstrukte unerheblich ist, ob die betrachteten Integrationsvorhaben bereits längere Zeit zurück liegen, so lange die Untersuchungsteilnehmer der Ansicht waren, die Charakteristika der Projekte noch angemessen in Erinnerung zu haben. Dennoch erscheint es gerade für den Bereich der Anwendungsintegration angesichts der dynamischen Weiterentwicklung der verfügbaren Technologien, Integrationsprodukte und -werkzeuge interessant, wann die betrachteten Integrationsvorhaben durchgeführt wurden.

Die Betrachtung der Jahre, in denen die Integrationsvorhaben abgeschlossen wurden, zeigt, dass einige der Projekte bereits 5 bis 10 Jahre zurückliegen, dass der überwiegende Anteil der Projekte jedoch näher an der Gegenwart liegt. Nur wenige der betrachteten Integrationsvorhaben waren zum Zeitpunkt der Interviews noch nicht abgeschlossen.

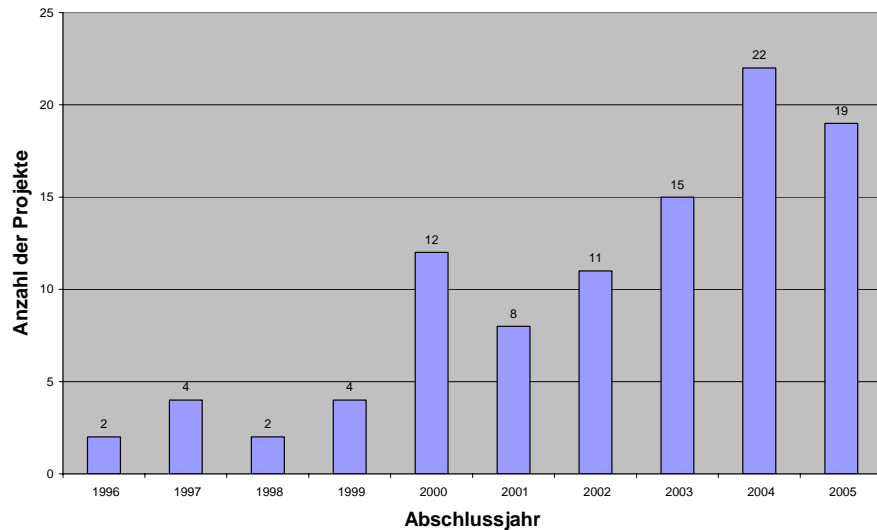


Abbildung 5-6: Anzahl der betrachteten Integrationsvorhaben, die in einem bestimmten Jahr abgeschlossen wurden

▪ **Zuständigkeit der Untersuchungsteilnehmer**

Um zu verstehen, welche Perspektive die Untersuchungsteilnehmer auf die betrachteten Integrationsvorhaben haben und welchen Einblick sie in die Charakteristika der Projekte hatten, wurden auch die jeweiligen Zuständigkeiten der Untersuchungsteilnehmer erfragt. Diesbezüglich muss beachtet werden, dass ein Untersuchungsteilnehmer in einzelnen Integrationsvorhaben verschiedene Zuständigkeiten innehaben konnte.

In 56 % der Fälle hatten die Untersuchungsteilnehmer die Projektleitung oder die Teilprojektleitung inne. In 41 % der Projekte waren sie als Architekt beteiligt. Ferner wirkten sie in 26 % der Projekte als Entwickler an der Implementierung der entworfenen Integrationslösung mit. Für einige der Integrationsvorhaben hatten die Untersuchungsteilnehmer ausschließlich die Planung, Aufwandschätzung oder Kalkulation durchgeführt (4 %), darüber hinaus besaßen sie in 11 % der Projekte sonstige Zuständigkeiten.

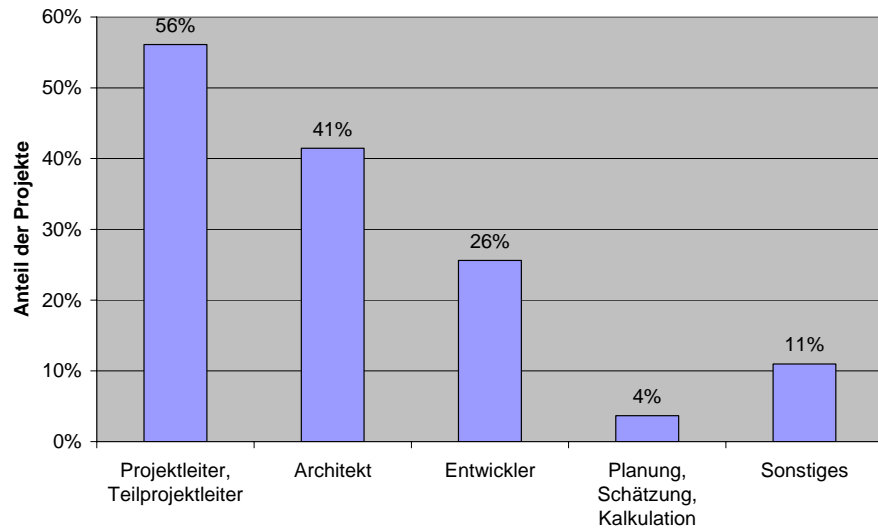


Abbildung 5-7: Zuständigkeit der Untersuchungsteilnehmer in den betrachteten Integrationsvorhaben

▪ Gegenstand der Integration

Wie erläutert, wurde durch die Untersuchungsteilnehmer anhand von fünfstufigen bipolaren Rating-Skalen beurteilt, inwiefern in den Integrationsvorhaben eine informationsorientierte, eine funktions- bzw. serviceorientierte oder eine geschäftsprozessorientierte Anwendungsintegration angestrebt wurde.

Die Ergebnisse weisen darauf hin, dass vor allem der Austausch von Informationen zwischen den Anwendungssystemen angestrebt wurde, gefolgt von der Verfügbarkeit von Funktionalität und der Automatisierung von Geschäftsprozessen. Diese Verteilung entspricht den Beschreibungen der Fachliteratur, gemäß denen die informationsorientierte Anwendungsintegration die traditionell überwiegende Integrationsperspektive darstellt, der ein Trend hin zu funktions- bzw. serviceorientierten und geschäftsprozessorientierten Integrationsvorhaben gegenüber gestellt wird.⁵⁵⁵

⁵⁵⁵ Vgl. beispielsweise Linthicum /Application Integration/ 4f.

- Item zur informationsorientierten Anwendungsintegration: ‚Das Projekt zielte insbesondere darauf, Informationen zwischen den betroffenen Anwendungssystemen auszutauschen.‘

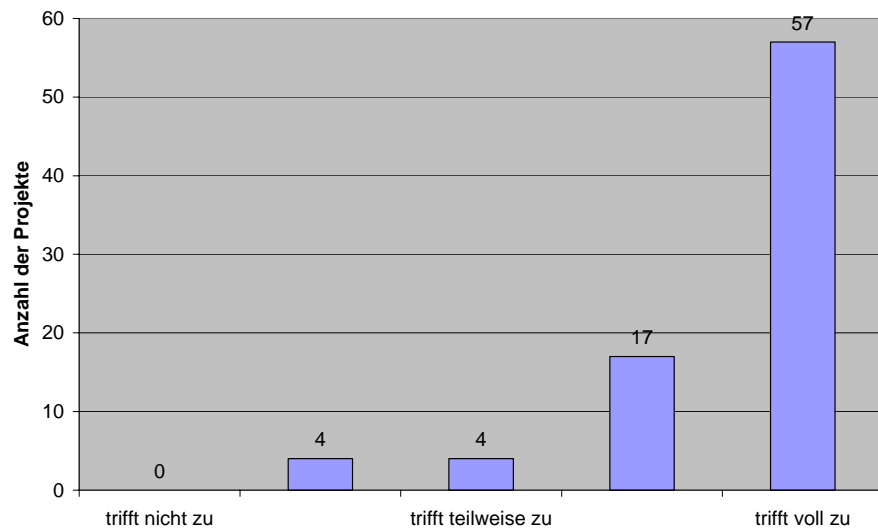


Abbildung 5-8: Beurteilung der Informationsorientierung der Integrationsvorhaben

- Item zur funktions- bzw. serviceorientierten Anwendungsintegration: ‚Das Projekt ziele insbesondere darauf, einzelne Funktionen der betroffenen Anwendungssysteme verfügbar zu machen (z. B. als Services) und sie flexibel zur neuartigen Unterstützung von Anwendungsgebieten nutzen zu können.‘

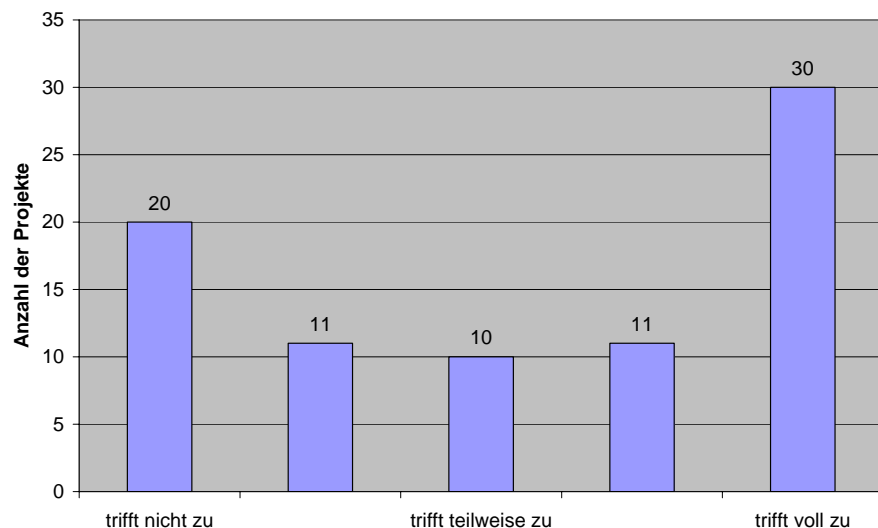


Abbildung 5-9: Beurteilung der Funktions- bzw. Serviceorientierung der Integrationsvorhaben

- Item zur geschäftsprozessorientierten Anwendungsintegration: ‚Das Projekt zielte insbesondere darauf, mehrstufige automatisierte Interaktionsfolgen zwischen den betroffenen Anwendungssystemen zur Abwicklung von Geschäftsprozessen zu realisieren.‘

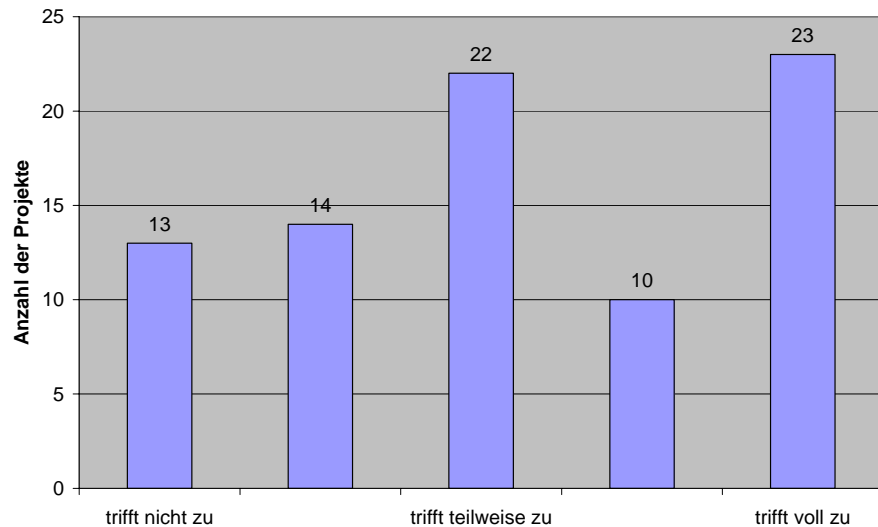


Abbildung 5-10: Beurteilung der Geschäftsprozessorientierung der Integrationsvorhaben

▪ Integrationsreichweite

Die Mehrzahl der betrachteten Integrationsvorhaben erforderten gemäß den Angaben der Untersuchungsteilnehmer nicht nur die Überbrückung von technischen, sondern auch die von organisatorischen Grenzen. Nur 12 % der Projekte betrafen die Integration von Anwendungssystemen innerhalb einer Abteilung eines Unternehmens (intra-Abteilung). 36 % der Projekte bezogen sich auf die Anwendungssysteme verschiedener Abteilungen eines Unternehmens (inter-Abteilung). 38 % der Projekte betrafen die Integration von Anwendungssystemen, die in verschiedenen Teilgesellschaften eines Konzerns betrieben wurden (intra-Konzern). 14 % der Projekte schließlich bezogen sich auf Anwendungssysteme mehrerer selbständiger Unternehmen (inter-Unternehmen).

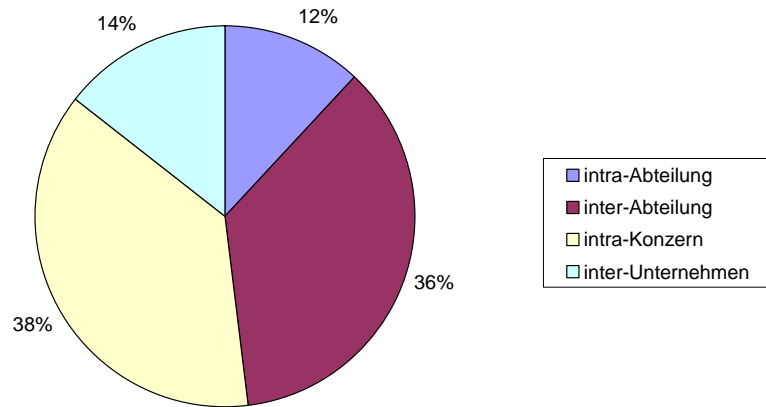


Abbildung 5-11: Integrationsreichweite in den betrachteten Integrationsvorhaben

▪ Die Internationalität der Integrationsvorhaben

Die Untersuchungsteilnehmer gaben des Weiteren an, dass sich die betrachteten Integrationsvorhaben überwiegend auf Anwendungssysteme von Organisationseinheiten innerhalb eines Landes bezogen (national). Knapp ein Drittel der Integrationsvorhaben wurde dagegen als international charakterisiert.

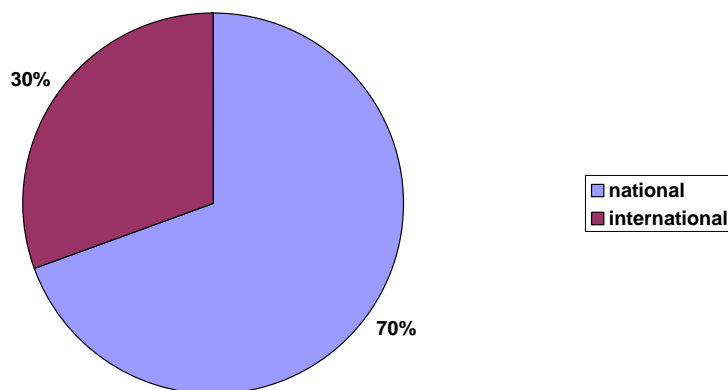


Abbildung 5-12: Internationalität der betrachteten Integrationsvorhaben

▪ Arten der integrierten Anwendungssysteme

Integrationsvorhaben werden durch die Anwendungssysteme geprägt, die zusammengeführt werden sollen. In den betrachteten Integrationsvorhaben waren zu 49 % ERP-Systeme, zu 60 % Altsysteme, zu 54 % sonstige standardisierte Anwendungssysteme (sonstige Standardanwendungen), zu 79 % sonstige individuell entwickelte Anwendungssysteme (sonstige Individualanwendungen) und zu 48 % internetbasierte Anwendungen betroffen. Diese Zahlen verdeutlichen, dass in den Integrationsvorhaben häufig verschiedene dieser Arten von Anwendungssystemen zusammengeführt wurden.

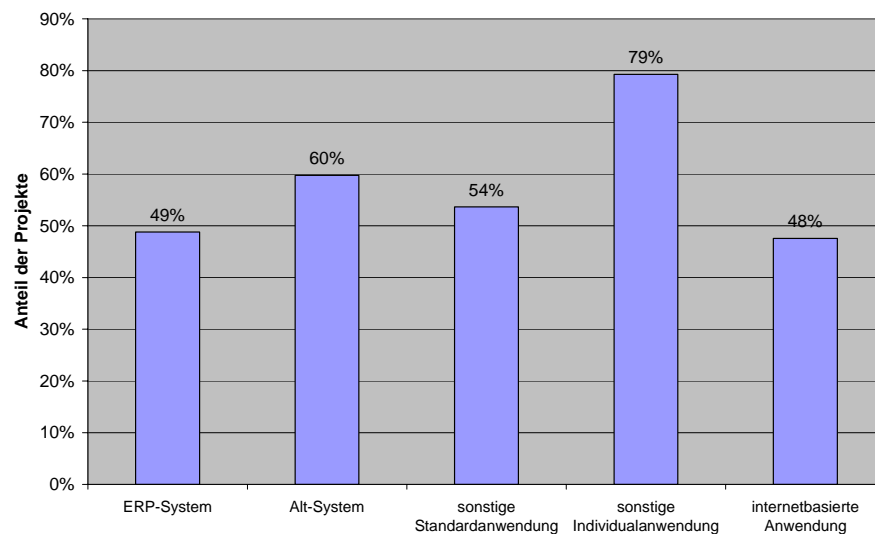


Abbildung 5-13: Die in den betrachteten Integrationsvorhaben zu integrierenden Anwendungssysteme

Darüber hinaus haben die Untersuchungsteilnehmer die Anzahl der zu verknüpfenden Anwendungssysteme eingeschätzt. Demnach wurden in den betrachteten Integrationsvorhaben durchschnittlich 18 Anwendungssysteme zusammengeführt. In 45 % der Projekte galt es jedoch lediglich bis zu 5 Anwendungssysteme zu integrieren.

▪ Die Neuartigkeit der Integrationslösung

Die Untersuchungsteilnehmer gaben schließlich an, inwiefern bei der Realisierung des angestrebten Integrationszustands auf bereits bestehenden Integrationsmaßnahmen aufgebaut werden konnte. Sie sollten dazu zwischen drei alternativen Ausprägungen auswählen, die als Stufen einer zunehmenden Wiederverwendung früherer Integrationsmaßnahmen verstanden werden können:

- ‚Im Rahmen des Projekts wurde eine vollständig neue Integrationsarchitektur implementiert‘ (Neuentwicklung).
- ‚Im Rahmen des Projekts wurde eine bestehende Integrationsarchitektur modifiziert oder erweitert‘ (Weiterentwicklung).
- ‚Im Rahmen des Projekts wurden weitere Anwendungssysteme an eine bestehende Integrationsinfrastruktur angebunden oder neue Interaktionen zwischen bereits angebundenen Anwendungssystemen realisiert‘ (Pflege).

Die Ergebnisse machen deutlich, dass in den meisten der betrachteten Integrationsvorhaben eine vollständig neue Integrationsarchitektur geschaffen wurde (67 %). In 25 % der Integrationsvorhaben wurde eine bestehende Integrationsarchitektur modifiziert oder erweitert, wogegen in 10 % der Projekte lediglich weitere Anwendungssysteme an eine bereits bestehende Integrationsinfrastruktur angebunden oder neue Interaktionen zwischen bereits angebundenen Anwendungssystemen realisiert wurden.

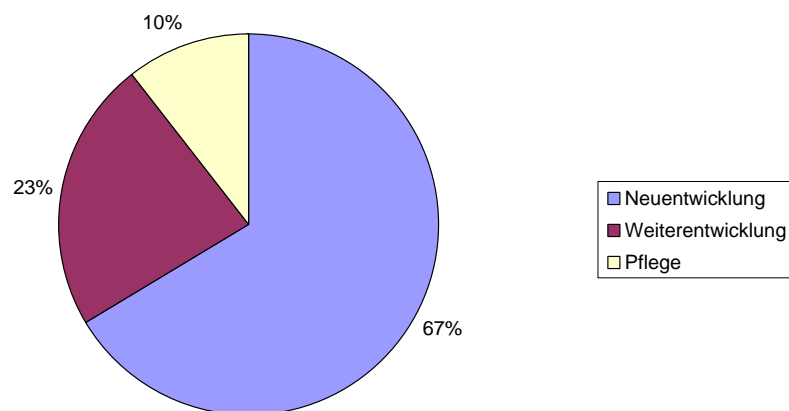


Abbildung 5-14: Neuartigkeit der Integrationslösungen in den betrachteten Integrationsvorhaben

5.3 Vorgehen zur inhaltlichen Analyse der erhobenen Konstrukte

In den 22 Interviews wurden insgesamt 422 persönliche Konstrukte auf Konstruktkarten notiert, im Durchschnitt wurden damit ca. 19 Konstrukte pro Interview erfasst. Die höchste Anzahl der in einem Interview erhobenen Konstrukte beträgt 31, die niedrigste Anzahl beträgt 11. Der Median der Anzahl der in den Interviews erhobenen Konstrukte liegt bei 17.

Es ist das Ziel der Analyse, die erhobenen Konstrukte unter inhaltlichen Gesichtspunkten zu strukturieren, um Gemeinsamkeiten und inhaltliche Schwerpunkte der Interpretationsmuster der Untersuchungsteilnehmer zu ermitteln. Auf diese Weise soll ein Gesamteindruck davon gewonnen werden, welche Aspekte der verglichenen Integrationsvorhaben von den Untersuchungsteilnehmern zur Erklärung des jeweiligen Entwicklungsaufwands angeführt wurden, d. h. welche Einflussfaktoren gemäß ihren persönlichen Erfahrungen im Kontext der Anwendungsintegration von Bedeutung sind. Da die Konstrukte im Wortlaut der Untersuchungsteilnehmer dokumentiert wurden, weisen sie im Detail eine große Vielfalt auf. Diese Reichhaltigkeit der Informationen soll möglichst erhalten werden.

Vor dem Hintergrund dieser Ziele erfolgte die inhaltliche Analyse der Konstrukte in drei Schritten: Es wurden mehrfach genannte Konstrukte identifiziert, die Konstrukte wurden aufgrund inhaltlicher Gemeinsamkeiten in Gruppen zusammengefasst und diese Gruppen wurden schließlich auf hohem Niveau in Themenbereiche eingeteilt. Da die in den Gesprächen notierten Bezeichnungen der Konstruktpole auf den teilweise unterschiedlichen Begriffsverständnissen der Auskunftspersonen basieren, wurde zudem eine begriffliche Harmonisierung angestrebt.

5.3.1 Bestimmung der Häufigkeiten der Konstrukte

Das Zählen der Häufigkeiten der Konstrukte wird von Stewart, Stewart und Fonda als grundlegender Schritt einer inhaltlichen Analyse erhobener Konstrukte dargestellt.⁵⁵⁶ Daher wurden im ersten Schritt der inhaltlichen Analyse übereinstimmende Konstrukte identifiziert.

⁵⁵⁶ Vgl. Stewart, Stewart, Fonda /Repertory Grid/ 47f.

Lediglich in solchen Fällen, bei denen die inhaltliche Deckungsgleichheit der erhobenen Konstrukte verschiedener Auskunftspersonen aufgrund der auf den Konstruktkarten notierten Formulierungen und unter Berücksichtigung der Transkripte mit großer Sicherheit gegeben schien, wurden sie als übereinstimmend gewertet.

In einigen Fällen schienen einzelne Konstrukte Teilmengen eines anderen Konstrukts abzubilden. Dies bedeutet, dass sie Merkmale von Integrationsvorhaben einzeln beschrieben, die in anderen Konstrukten gemeinsam erfasst worden waren.⁵⁵⁷ Dabei beschrieben sie die Merkmale der Integrationsvorhaben auf demselben Abstraktionsniveau, so dass sie nicht als über- bzw. untergeordnete Konstrukte in den Konstruktsystemen der Auskunftspersonen, sondern als nebeneinander stehend angesehen werden können. Nur wenn aufgrund der Transkripte sicher schien, dass die Verbindung der Merkmale nicht als grundlegend für die umfassenderen Konstrukte angesehen werden muss, wurden sie als übereinstimmend mit den einzelnen Konstrukten geringeren Umfangs interpretiert, d. h. in die einzelnen beschriebenen Merkmale aufgeteilt. Dieses Vorgehen betraf insbesondere Konstrukte zu den Anforderungen an die qualitätsbezogenen Merkmale einer Integrationslösung. Beispielsweise bezog sich ein Konstrukt auf die Anforderungen an die Sicherheit, ein weiteres Konstrukt auf die Anforderungen an die Verfügbarkeit, während sich ein drittes Konstrukt sowohl auf die Anforderungen an die Sicherheit als auch auf die Anforderungen an die Verfügbarkeit bezog.

Infolge der Berücksichtigung dieser Überschneidungen wurden die 422 beschrifteten Konstruktkarten auf 310 unterschiedliche Konstrukte reduziert. Insgesamt wurden 64 Konstrukte identifiziert, die mit großer Genauigkeit von mehreren Auskunftspersonen geteilt werden. Die höchste Anzahl der Nennungen wies ein Konstrukt auf, das in 8 Interviews erhoben wurde und die räumliche Trennung der Projektbeteiligten betrifft. In den meisten Fällen wurden die mehrfach genannten Konstrukte jedoch nur in 2 Interviews erhoben. Im Durchschnitt wurden die einzelnen Konstrukte damit 1,4-mal genannt.

⁵⁵⁷ Im Weiteren wird vereinfachend davon gesprochen, dass die erhobenen Konstrukte Merkmale von Integrationsvorhaben beschreiben. Dieses Verständnis der Konstrukte wird jedoch in Kapitel 5.2.2.2 präzisiert, wonach die Konstrukte jeweils zwei entgegengesetzte Ausprägungen von Merkmalen bestimmter Objekte in Integrationsvorhaben beschreiben.

Es existieren nur wenige Orientierungspunkte für die Interpretation dieser festgestellten Häufigkeiten der erhobenen Konstrukte. In dem bereits mehrfach angesprochenen Beitrag von Moynihan werden beispielsweise 113 Konstrukte von 14 Auskunftspersonen verschiedener Unternehmen erfasst, die im Durchschnitt 1,8-mal genannt wurden.⁵⁵⁸

Whyte und Bytheway beschreiben des Weiteren die Erhebung persönlicher Konstrukte von Anwendern, um deren Verständnis von den Erfolgsfaktoren von Informationssystemen zu erfassen. In Gesprächen mit 10 Auskunftspersonen eines Unternehmens werden 43 Konstrukte erhoben, die im Durchschnitt 2,3-mal genannt wurden.⁵⁵⁹

Die in der vorliegenden Untersuchung festgestellten Häufigkeiten der Konstrukte erscheinen im Vergleich dazu relativ gering. Dies könnte auf verschiedene Ursachen zurückzuführen sein. Zum einen wurde in der vorliegenden Untersuchung eine höhere Anzahl von Auskunftspersonen befragt, die im Unterschied zur Studie von Whyte und Bytheway aus verschiedenen Unternehmen stammen. Zum anderen wurden die Konstrukte im Rahmen der Interviews im Wortlaut der Teilnehmer notiert, während Moynihan die nachträgliche Identifizierung der Konstrukte durch die Wissenschaftler anhand von Gesprächsaufzeichnungen beschreibt. Schließlich kann die Vielfalt der in der vorliegenden Untersuchung erhobenen Konstrukte ein Indiz für die Komplexität der betrachteten Wissensdomäne sein. Dies scheint die Vermutung zu bestätigen, dass von einer großen Anzahl interdependenter Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand auszugehen ist.

5.3.2 Begriffliche Harmonisierung der Konstrukte

Um ein einheitliches Begriffssystem und eine hohe Verständlichkeit der Konstrukte zu erreichen, wurden die Formulierungen der Konstruktpole teilweise geringfügig angepasst. Dies war vor allem im Zusammenhang mit den identifizierten Überschneidungen erforderlich. Wiederum wurde anhand der Transkripte Sorge getragen, die Aussagen der Untersuchungsteilnehmer nicht inhaltlich zu verfremden.

Beispielsweise verwendeten einige Teilnehmer die Bezeichnung „Integrationstool“ und andere Teilnehmer die Bezeichnung „Integrationsprodukt“. In beiden Fällen bezogen sich die Praktiker auf Integrationsprodukte im Sinne dieser Arbeit. Im Zuge der begriff-

⁵⁵⁸ Vgl. Moynihan /Project Risk/ 360f.

⁵⁵⁹ Vgl. Whyte, Bytheway /Information System Success/ 80-82.

lichen Harmonisierung wurde daher stets die Bezeichnung „Integrationsprodukt“ eingesetzt.

Im Anschluss an diese Maßnahmen wurde jedem Untersuchungsteilnehmer eine Liste der von ihm genannten Konstrukte zugesandt und dabei die ursprünglichen Formulierungen den überarbeiteten Formulierungen gegenübergestellt. Die Teilnehmer wurden gebeten, zu überprüfen, ob die überarbeiteten Formulierungen und in Einzelfällen die zuvor erläuterte Aufspaltung zusammen genannter Sachverhalte in einzelne Konstrukte in ihrem Sinne sind, d. h. dass sie ihre Erfahrungen und Ansichten weiterhin akkurat wiedergeben. In wenigen Fällen wurden daraufhin die vorgeschlagenen Formulierungen auf Wunsch der Untersuchungsteilnehmer nochmals modifiziert.

Dieses Vorgehen entspricht den in der Fachliteratur gestellten Forderungen zur Sicherstellung der Validität der Ergebnisse.⁵⁶⁰ Es wird daher davon ausgegangen, dass die im Weiteren verwendeten Formulierungen der Konstrukte die Erfahrungen und Ansichten der Auskunftspersonen präzise wiedergeben.

5.3.3 Einteilung der erhobenen Konstrukte in Gruppen und Themenbereiche

Die Aufdeckung der inhaltlichen Gemeinsamkeiten verschiedener Konstrukte basiert im Falle qualitativer Forschungsansätze auf den Urteilen der an der Analyse beteiligten Personen.⁵⁶¹ Im Unterschied zu den formal-mathematischen Qualitätskriterien quantitativer Analysen, wird die Güte qualitativer Analysen insbesondere auf zwei Kriterien zurückgeführt: Intersubjektivität und die Validierung der Analyseergebnisse durch die Auskunftspersonen.⁵⁶² An diesen Kriterien wurde auch der anschließende Schritt der inhaltlichen Analyse der Untersuchungsergebnisse ausgerichtet: die Gruppierung der erhobenen Konstrukte und die Ermittlung der inhaltlichen Schwerpunkte im Verständnis der erfahrenen Praktiker von den Einflussfaktoren auf den Entwicklungsaufwand.

⁵⁶⁰ Nelson u. a. /Software Operations Support Expertise/ 483.

⁵⁶¹ Vgl. Whyte, Bytheway /Information System Success/ 83. Im Falle quantitativer Forschungsansätze können dagegen die Korrelationen zwischen den Konstrukten anhand der Verbindungen zwischen Elementen und Konstrukten statistisch berechnet werden (vgl. beispielsweise Stewart, Stewart, Fonda /Repertory Grid/ 56-66).

⁵⁶² Vgl. Stewart, Stewart, Fonda /Repertory Grid/ 50f.; Whyte, Bytheway /Information System Success/ 83.

Für die Gruppierung von Konstrukten kann ein von oben nach unten gerichtetes Vorgehen (top down) gewählt werden. Dabei wird vorab ein System von Kategorien gebildet, denen anschließend die erhobenen Konstrukte zugeordnet werden. Für die vorliegende Untersuchung erscheint diese Vorgehensweise jedoch angesichts der verfolgten Explorationsstrategie nicht als geeignet. Stattdessen wurde ein von unten nach oben gerichtetes Vorgehen (bottom up) gewählt, bei dem die Konstrukte aufgrund inhaltlicher Gesichtspunkte zunächst zu Gruppen unterschiedlicher Größe zusammengefasst wurden. Dabei sollte die Vielfalt der in den Konstrukten erfassten Informationen erhalten werden. Diesen Gruppen wurden anschließend Bezeichnungen zugeordnet, die den gemeinsamen Gegenstand der Konstrukte erfassen sollen. Die abgegrenzten Gruppen von Konstrukten und die durch sie beschriebenen Attribute von Integrationsvorhaben wurden in einem weiteren Schritt in verschiedene Themenbereiche eingeteilt. Nachdem in den vorangehenden Schritten angestrebt wurde, die Vielfalt der Informationen zu bewahren, wurden nunmehr bewusst umfangreiche Bereiche gebildet. Der Alternative, weitere zwischengeschaltete Aggregationsstufen zu bilden, wurde kein wesentlicher Mehrwert zugeordnet.⁵⁶³

Wiederum wurden die Transkripte der Interviews hinzugezogen, um die Konstrukte stets im Sinne der Auskunftspersonen zu interpretieren. Die auf diese Weise vorgenommene Gruppierung der Konstrukte, die Abgrenzung der Themenbereiche und die jeweils gewählten Bezeichnungen wurden anschließend durch verschiedene akademische Mitarbeiter überprüft, um eine intersubjektive Plausibilisierung der inhaltlichen Strukturierung der Konstrukte zu erreichen. In einigen Fällen wurden die Zuordnungen oder die Bezeichnungen darauf hin angepasst. Schließlich wurden die Ergebnisse der inhaltlichen Analyse an die Untersuchungsteilnehmer versandt. Dabei wurden stets die durch den jeweiligen Teilnehmer genannten Konstrukte markiert und darum gebeten, ihre Zuordnung zu überprüfen und ggf. Änderungswünsche zu äußern. Da die Untersuchungsteilnehmer keinerlei Einwände gegenüber der vorgenommenen Strukturierung der Konstrukte vorbrachten, wird davon ausgegangen, dass die Ergebnisse der inhaltlichen Analyse ihre Erfahrungen und Ansichten akkurat wiedergeben.⁵⁶⁴

⁵⁶³ Zur inhaltlichen Gruppierung von Konstrukten vgl. beispielsweise Stewart, Stewart, Fonda /Repertory Grid/ 48-51.

⁵⁶⁴ Dieses Vorgehen entspricht beispielsweise den Empfehlungen von Stewart, Stewart und Fonda (vgl. Stewart, Stewart, Fonda /Repertory Grid/ 51) bzw. der von Whyte und Bytheway beschriebenen Vorgehensweise (vgl. Whyte, Bytheway /Information System Success/ 83). Moynihan geht dagegen

Auf diese Weise wurden die 310 erfassten Konstrukte in 91 Gruppen eingeteilt. Die durchschnittliche Anzahl der Konstrukte einer Gruppe beträgt damit ca. 3. Die Gruppe mit dem größten Umfang beinhaltet 11 Konstrukte. Die meisten Gruppen umfassen lediglich 2 persönliche Konstrukte. Bei 13 Konstrukten herrschte Unsicherheit, mit welchen anderen Konstrukten sie eine hinreichende inhaltliche Ähnlichkeit aufweisen, um sie zu einer Gruppe zusammenzufassen. Sie wurden daher als eigenständige Konstrukte belassen, d. h. sie beschreiben alleine ein bestimmtes Merkmal von Integrationsvorhaben.

Wiederum können die von Moynihan sowie die von Whyte und Bytheway beschriebenen Studien als Vergleichspunkte hinzugezogen werden. Während Moynihan die 113 erhobenen Konstrukte in 22 thematische Gruppen eingeteilt werden (durchschnittlich ca. 5 Konstrukte pro Gruppe), wird bei Whyte und Bytheway aus den 43 erhobenen Konstrukten auf 21 Attribute geschlossen (durchschnittlich ca. 2 Konstrukte pro Gruppe).⁵⁶⁵ Die in der vorliegenden Untersuchung vorgenommene Verdichtung der Konstrukte liegt mit einer durchschnittlichen Gruppengröße von ca. 3 Konstrukten zwischen den beiden referenzierten Beiträgen.

Die 91 Gruppen der Konstrukte wurden anschließend in 9 Themenbereiche eingeteilt. Die durchschnittliche Anzahl der Gruppen eines Themenbereichs beträgt damit ca. 10. Die durchschnittliche Anzahl der Konstrukte eines Themenbereichs beträgt 34. Die beiden umfangreichsten Themenbereiche sind *die Anforderungen an die Integrationslösung* (55 Konstrukte in 15 Gruppen) und *die Gestaltung der Integrationslösung* (49 Konstrukte in 19 Gruppen).

Die vorgenommene Gruppierung der Konstrukte und die Abgrenzung der Themenbereiche können jedoch nicht den Anspruch erheben, die alleingültige oder bestmögliche Weise der inhaltlichen Strukturierung der erhobenen Konstrukte darzustellen. Sie repräsentiert lediglich die Urteile der an der Analyse beteiligten Personen. Darüber hinaus sind andere Kriterien für die Organisation der Konstrukte denkbar, beispielsweise in-

nicht auf Maßnahmen zur intersubjektiven Plausibilisierung der vorgenommenen Gruppierung ein (vgl. Moynihan /Project Risk/ 361).

⁵⁶⁵ Vgl. Moynihan /Project Risk/ 365-371; Whyte, Bytheway /Information System Success/ 84.

wiefern sie durch die Untersuchungsteilnehmer als charakteristisch für die Anwendungsintegration angesehen werden, welche Bedeutung für den Entwicklungsaufwand ihnen durch die Untersuchungsteilnehmer zugemessen wird, zu welchem Zeitpunkt im Projektverlauf sie eingeschätzt werden können oder inwiefern sie durch die Projektleitung beeinflusst werden können oder nicht.

5.3.4 Trendanalyse der inhaltlichen Gemeinsamkeiten der erhobenen Konstrukte

Im Zusammenhang mit der Gestaltung des Auswahlverfahrens der Auskunftspersonen wurde erläutert, dass die Neuartigkeit der mit jedem weiteren Interview erhobenen Konstrukte als Indikator dafür angesehen wird, inwiefern das Verständnis einer Domäne angemessen abgebildet werden konnte.⁵⁶⁶ Nach Abschluss der inhaltlichen Analyse der erhobenen Konstrukte wurde daher überprüft, wie die ermittelten Gruppen auf die durchgeführten Interviews zurückgeführt werden können. Es stellte sich die Frage, ob ein Trend bezüglich der inhaltlichen Gemeinsamkeiten im Zuge der Interviews festzustellen ist und insbesondere, ob weitere Auskunftspersonen in die Untersuchung einbezogen werden sollten.

In Tabelle 5-2 ist die kumulierte Anzahl der Gruppen dargestellt, denen die im Zuge der durchgeführten Interviews erhobenen Konstrukte zugeordnet wurden. Trotz der geringen durchschnittlichen Gruppengröße zeigt sich, dass mit den letzten drei durchgeführten Interviews keine Konstrukte zu neuen Attributen von Integrationsvorhaben erhoben wurden.

Anzahl der Interviews	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Kumulierte Anzahl der Gruppen	9	20	27	43	51	56	60	63	66	69	72	72	75	78	78	81	84	89	91	91	91	91

Tabelle 5-2: Kumulierte Anzahl der ermittelten Gruppen in Abhängigkeit von der Anzahl der durchgeführten Interviews

⁵⁶⁶ Vgl. Moynihan /Project Risk/ 362; Tan, Hunter /Repertory Grid/ 50 bzw. Kapitel 4.3.4.

Der Trend der inhaltlichen Gemeinsamkeiten im Zuge der Interviews wird in Abbildung 5-15 graphisch dargestellt. Es zeigt sich eine degressive Entwicklung des kumulierten Anteils der ermittelten Gruppen im Verhältnis zum kumulierten Anteil der durchgeführten Interviews.

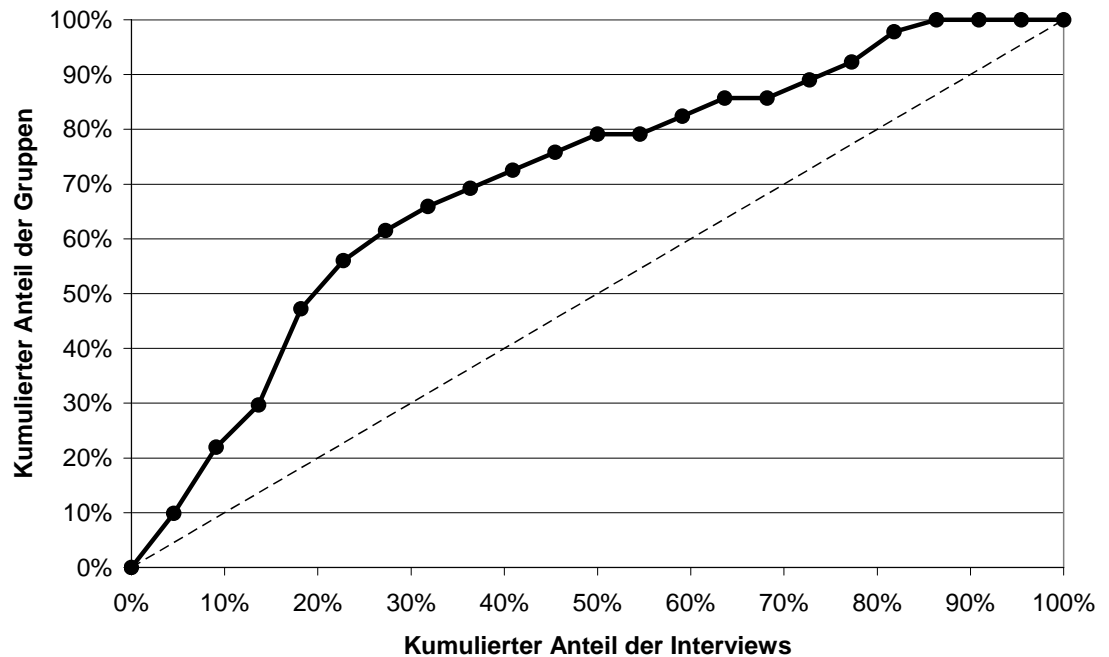


Abbildung 5-15: Kumulierter Anteil der ermittelten Gruppen in Abhängigkeit vom kumulierten Anteil der durchgeführten Interviews

Aufgrund dieser Entwicklung wurde entschieden, keine weiteren Auskunftspersonen in die Erhebung einzubeziehen. Es ist zwar davon auszugehen, dass durch weitere Untersuchungsteilnehmer neuen Konstrukte erhoben werden könnten, jedoch erscheint es sehr wahrscheinlich, dass diese Konstrukte große inhaltliche Gemeinsamkeiten mit den bereits erhobenen Interpretationsmustern aufweisen würden.⁵⁶⁷

5.4 Ergebnisse der inhaltlichen Analyse der erhobenen Konstrukte

Im Weiteren werden die Ergebnisse der inhaltlichen Analyse dargestellt. Im Sinne der Übersichtlichkeit wird dabei eine Vorgehensweise gewählt, die dem beschriebenen Vorgehen bei der Analyse der Konstrukte entgegengesetzt ist. Anstelle von den einzel-

⁵⁶⁷ In gleicher Weise begründet Moynihan den Abschluss der Konstrukterhebung bereits nach 14 Interviews (vgl. Moynihan /Project Risk/ 361f.).

nen Konstrukten auszugehen und diese zu Gruppen und Themenbereichen zusammenzufassen wird somit mit der Darstellung der abgegrenzten Themenbereiche begonnen, um schrittweise zu den einzelnen Konstrukten vorzugehen. Dabei wird im Interesse der Übersichtlichkeit eine tabellarische Darstellungsform der erhobenen Konstrukte gewählt.

Um einen Überblick über die Beurteilung der erhobenen Konstrukte durch die Untersuchungsteilnehmer zu erhalten, wird jedoch zunächst deren Kommentierung zusammenfassend dargestellt.

5.4.1 Kommentierung der Konstrukte durch die Untersuchungsteilnehmer

Die Kommentierung der Konstrukte durch die Untersuchungsteilnehmer betraf insbesondere die Beurteilung ihrer Bedeutung für den Entwicklungsaufwand sowie ihre Einschätzung als Charakteristikum von Integrationsvorhaben.

▪ Beurteilung der Bedeutung für den Entwicklungsaufwand

Die Untersuchungsteilnehmer wurden im Rahmen der Interviews gebeten, die Konstrukte hinsichtlich der Bedeutung der auf ihnen beschriebenen Merkmale für den Entwicklungsaufwand in drei Gruppen einzuteilen. Auf diese Weise wurde 128 Konstrukten (41 %) zumindest durch einen Untersuchungsteilnehmer eine hohe Bedeutung, 148 Konstrukten (48 %) zumindest durch einen Untersuchungsteilnehmer eine mittlere Bedeutung und 89 Konstrukten (29 %) zumindest durch einen Untersuchungsteilnehmer eine niedrige Bedeutung für den Entwicklungsaufwand zugemessen. Diese Beurteilungen sprechen dafür, dass in der Untersuchung schwerpunktmäßig solche Konstrukte erfasst wurden, die aus Sicht der Teilnehmer maßgebliche Einflussfaktoren auf den Entwicklungsaufwand abbilden.

Vorgegebene Kategorien	Anzahl (Anteil) der zugeordneten Konstrukte	
Das Konstrukt beschreibt ein Merkmal, das von hoher Bedeutung für den Entwicklungsaufwand ist.	128	(41 %)
Das Konstrukt beschreibt ein Merkmal, das von mittlerer Bedeutung für den Entwicklungsaufwand ist.	148	(48 %)
Das Konstrukt beschreibt ein Merkmal, das von niedriger Bedeutung für den Entwicklungsaufwand ist.	89	(29 %)

Tabelle 5-3: Beurteilung der Bedeutung der erhobenen Konstrukte für den Entwicklungsaufwand

Wie erläutert sollte der Einfluss der beschriebenen Merkmale auf den Entwicklungsaufwand bei der Einteilung der Konstrukte isoliert voneinander und unter Bezug auf den markierten aufwandstreibenden Konstruktpol eingeschätzt werden. Die Untersuchungsteilnehmer wiesen jedoch oftmals darauf hin, dass die Bedeutung des durch den Konstruktpol beschriebenen Sachverhalts für den Entwicklungsaufwand variieren kann. Aus ihrer Sicht bestehen zudem häufig komplizierte Wechselwirkungen zwischen den auf verschiedenen Konstruktkarten notierten Merkmalen. Sie begründeten damit in vielen Fällen die Einordnung eines Konstrukts in die Kategorie der mittleren Bedeutung für den Entwicklungsaufwand, obwohl das beschriebene Merkmal unter bestimmten Bedingungen auch eine große bzw. niedrige Bedeutung besitzen könne.

Des Weiteren wurden Konstrukte, die von mehreren Untersuchungsteilnehmern geteilt werden, von diesen teilweise unterschiedlich hinsichtlich ihrer Bedeutung beurteilt. Diese vermeintlichen Widersprüche reflektieren zum einen die unterschiedliche Stellung der Konstrukte in den individuellen Konstruktsystemen der Untersuchungsteilnehmer und bringen damit Unterschiede im Verständnis der Teilnehmer von den maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand zum Ausdruck. Zum anderen muss die Kommentierung der Konstrukte durch die Untersuchungsteilnehmer im Kontext der weiteren Konstrukte interpretiert werden, die im Rahmen der betreffenden Interviews erhoben wurden. So dürfte die Einordnung der Konstrukte durch die Untersuchungsteilnehmer oftmals deren relative und nicht deren absolute Bedeutung ausdrücken.⁵⁶⁸

▪ **Einschätzung als Charakteristikum von Integrationsvorhaben**

Des Weiteren wurden die Untersuchungsteilnehmer im Rahmen der Interviews gebeten, diejenigen der von ihnen genannten Konstrukte auszuwählen, die Merkmale beschreiben, die sie als charakteristisch für Integrationsvorhaben ansehen. Auf diese Weise wurden 169 Konstrukte (55 %) zumindest durch einen Untersuchungsteilnehmer als Charakteristika von Integrationsvorhaben eingeschätzt. Dieses Ergebnis verdeutlicht,

⁵⁶⁸ Eine weiterführende mathematische Analyse der Einordnung der Konstrukte, zur Bestimmung der Häufigkeiten, mit denen einem Konstrukt höhere, gleiche oder geringere Bedeutung für den Entwicklungsaufwand im Vergleich zu anderen Konstrukten beigemessen wird, erscheint aufgrund der Vielfalt der erhobenen Informationen wenig aussagekräftig. Sie könnte vielmehr der Gegenstand anschließender Untersuchungen unter einem quantitativen bzw. normativen Forschungsansatz sein.

dass in der Untersuchung aus Sicht der Teilnehmer viele repräsentative Besonderheiten der Anwendungsintegration erfasst wurden.

Die Entscheidungen, ein Konstrukt nicht als Charakteristikum von Integrationsvorhaben zu bezeichnen, wurden durch die Untersuchungsteilnehmer insbesondere auf zwei Weisen begründet. Teilweise sehen die Untersuchungsteilnehmer Gemeinsamkeiten zwischen Integrations- und anderen Entwicklungsvorhaben bezüglich der betreffenden Einflussfaktoren auf den Entwicklungsaufwand. Teilweise erschienen den Untersuchungsteilnehmern die verglichenen Integrationsvorhaben hinsichtlich der in den Konstrukten beschriebenen Merkmale dagegen nicht als repräsentativ. Auch wenn es sich aus ihrer Sicht um Besonderheiten der Anwendungsintegration handelte, sahen sie sie dennoch nicht als charakteristisch (im Sinne von typisch) für Integrationsvorhaben an.

5.4.2 Die abgegrenzten Themenbereiche im Überblick

Im Zuge der inhaltlichen Analyse wurden die 310 erhobenen Konstrukte in 91 Gruppen eingeteilt, die anschließend zu 9 Themenbereichen unterschiedlichen Umfangs zusammengefasst wurden.

Den umfangreichsten Themenbereich stellen *die Anforderungen an die Integrationslösung* dar. Diesem Bereich wurden 55 der erhobenen persönlichen Konstrukte zugeordnet (entspricht einem Anteil von ca. 18 %). Als weitere Themenbereiche wurden *die Charakteristika der zu integrierenden Anwendungssysteme* (35 Konstrukte, entsprechen 11 %), *die Gestaltung der Integrationslösung* (49 Konstrukte, entsprechen 16 %), *die eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge* (29 Konstrukte, entsprechen 9 %), *das Testen und die Überführung in die Nutzung* (17 Konstrukte, entsprechen 5 %), *die Kunden und Stakeholder der Integrationslösung* (34 Konstrukte, entsprechen 11 %), *die Projektmitarbeiter* (32 Konstrukte, entsprechen 10 %), *das Management des Integrationsvorhabens* (22 Konstrukte, entsprechen 7 %) und *die Rahmenbedingungen des Integrationsvorhabens* (37 Konstrukte, entsprechen 12 %) identifiziert.

In den einzelnen Themenbereichen sind mehrere Konstrukte derselben Untersuchungsteilnehmer enthalten. Der Umfang der Themenbereiche ist jedoch nicht lediglich auf

eine große inhaltliche Ähnlichkeit der Konstrukte weniger Untersuchungsteilnehmer zurückzuführen. Vielmehr umfassen die Themenbereiche im Allgemeinen die Konstrukte eines großen Anteils der Untersuchungsteilnehmer. Unter diesem Gesichtspunkt fällt der Themenbereich *der Projektmitarbeiter* auf, dem Konstrukte aller Untersuchungsteilnehmer zugeordnet wurden.

Die unterschiedenen Themenbereiche sind in Tabelle 5-4 dargestellt. Neben der gewählten Bezeichnung der Themenbereiche ist in der Tabelle angegeben, wie viele der persönlichen Konstrukte den einzelnen Themenbereichen zugeordnet wurden. Dabei sind sowohl die absolute Anzahl der zugeordneten Konstrukte als auch deren Anteil an den 310 insgesamt erhobenen Konstrukten angegeben. Anhand dieser Angaben wird der unterschiedliche Umfang der Themenbereiche deutlich. Darüber hinaus wird in der Tabelle dargestellt, von wie vielen Untersuchungsteilnehmern die Konstrukte eines Themenbereichs stammen.

Themenbereich	Anzahl (Anteil) der Konstrukte	Anzahl der Teilnehmer
Die Anforderungen an die Integrationslösung	55 (18 %)	20
Die Charakteristika der zu integrierenden Anwendungssysteme	35 (11 %)	20
Die Gestaltung der Integrationslösung	49 (16 %)	20
Die eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge	29 (9 %)	18
Das Testen der Integrationslösung und die Überleitung in die Nutzung	17 (5 %)	10
Der Kunde und die Stakeholder der Integrationslösung	34 (11 %)	19
Die Projektmitarbeiter	32 (10 %)	22
Das Management des Integrationsvorhabens	22 (7 %)	12
Die Rahmenbedingungen des Integrationsvorhabens	37 (12 %)	20

Tabelle 5-4: Unterschiedene Themenbereiche der erhobenen persönlichen Konstrukte

Im Zuge der Darstellung der bisherigen Erkenntnisse zu den Einflussfaktoren auf den Entwicklungsaufwand in der Softwareentwicklung wurde darauf hingewiesen, dass in der Fachliteratur ein breites Spektrum von Einflussfaktoren behandelt wird, das von konzeptionellen Merkmalen der entwickelten Software über die verwendeten Technologien zu den Projektmitarbeitern und den herrschenden Rahmenbedingungen reicht. Dabei werden sowohl Aspekte der Entwicklungs- und Unterstützungsaufgaben als auch

der organisatorischen Gestaltungsmaßnahmen thematisiert.⁵⁶⁹ Anhand der unterschiedenen Themenbereiche wird deutlich, dass sich diese Vielfalt der Einflussfaktoren auf den Entwicklungsaufwand auch in den Konstruktsystemen der Untersuchungsteilnehmer widerspiegelt.

Gleichwohl betreffen die einzelnen Themenbereiche aus Sicht der Teilnehmer nicht in gleicher Weise Besonderheiten der Anwendungsintegration (siehe Tabelle 5-5). Während der Anteil der Konstrukte, die zumindest durch einen Untersuchungsteilnehmer als charakteristisch angesehen werden, in den ersten vier der vorgestellten Themenbereiche von 60 % bis zu 81 % reicht, liegt er in den restlichen Themenbereichen lediglich zwischen 27 % und 44 %. Demnach sind insbesondere mit *den Anforderungen an die Integrationslösung*, *den Charakteristika der zu integrierenden Anwendungssysteme*, *der Gestaltung der Integrationslösung* und *den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen* Einflussfaktoren auf den Entwicklungsaufwand verbunden, die Integrationsvorhaben gegenüber anderen Entwicklungsvorhaben prägen. Angesichts des Zuschnitts dieser Themenbereiche und den gewählten Bezeichnungen erscheint der hohe Anteil als charakteristisch angesehener Sachverhalte nahe liegend. Es wird jedoch ersichtlich, dass auch mit solchen Themenbereichen, die vermeintlich allgemeingültige Merkmale von Projekten betreffen, wie beispielsweise *die Projektmitarbeiter* oder *die Kunden und Stakeholder*, ebenfalls Charakteristika von Integrationsvorhaben in Verbindung gebracht werden.

Auch die Einschätzung der in den Konstrukten beschriebenen Merkmale hinsichtlich ihrer Bedeutung für den Entwicklungsaufwand fällt in den Themenbereichen unterschiedlich aus. Der Anteil der Konstrukte, denen zumindest durch einen Untersuchungsteilnehmer eine hohe Bedeutung beigemessen wird, ist insbesondere im Zusammenhang mit *den Charakteristika der zu integrierenden Anwendungssysteme* und *den Anforderungen an die Integrationslösung* mit 57 % bzw. 53 % sehr hoch. Dagegen werden 62 % der Konstrukte zu *den Rahmenbedingungen des Integrationsvorhabens* zumindest durch einen Teilnehmer lediglich eine geringe Bedeutung zugestanden (siehe Tabelle 5-5).

Es muss jedoch generell beachtet werden, dass diese Urteile lediglich auf eine geringe Anzahl von Auskunftspersonen zurückzuführen sind. Wie erläutert wurden die Konstrukte ausschließlich in den Interviews kommentiert, in denen sie erhoben wurden. In vielen Fällen können daher nur die Einschätzungen einzelner Untersuchungsteilnehmer

⁵⁶⁹ Siehe Kapitel 3.1.2.

wiedergegeben werden. Zudem erfolgte die Kommentierung der einzelnen Konstrukte stets im spezifischen Kontext der weiteren Konstrukte, die in den betreffenden Interviews notiert worden waren. Die angegebenen Prozentwerte sollten daher ausschließlich im Sinne des verfolgten qualitativen und idiographischen Forschungsansatzes interpretiert werden: als Kommentierung der Konstrukte durch die Untersuchungsteilnehmer, als Ausdruck ihrer persönlichen Erfahrungswerte und Ansichten. Quantitative bzw. normative Schlüsse, insbesondere auf die tatsächliche Wirkungsstärke der beschriebenen Einflussfaktoren auf den Entwicklungsaufwand, sind dagegen nicht möglich.

Themenbereich	Anzahl der Konstrukte	Anteilige Einschätzung der Konstrukte			
		Charakteristika	Bedeutung für den Entwicklungsaufwand		
			hoch	mittel	niedrig
Die Anforderungen an die Integrationslösung	55	60 %	53 %	56 %	15 %
Die Charakteristika der zu integrierenden Anwendungssysteme	35	83 %	57 %	46 %	23 %
Die Gestaltung der Integrationslösung	49	78 %	41 %	45 %	31 %
Die eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge	29	66 %	38 %	59 %	24 %
Das Testen der Integrationslösung und die Überleitung in die Nutzung	17	29 %	29 %	47 %	29 %
Der Kunde/die Stakeholder der Integrationslösung	34	35 %	29 %	47 %	35 %
Die Projektmitarbeiter	32	44 %	44 %	44 %	22 %
Das Management des Integrationsvorhabens	22	27 %	36 %	41 %	23 %
Die Rahmenbedingungen des Integrationsvorhabens	37	38 %	30 %	38 %	62 %

Tabelle 5-5: Kommentierung der Konstrukte in den unterschiedenen Themenbereichen

5.4.3 Einflussfaktoren im Zusammenhang mit den Anforderungen an die Integrationslösung

Dem Themenbereich der Anforderungen an die Integrationslösung wurden 15 Gruppen zugeordnet, die insgesamt 55 der erhobenen Konstrukte umfassen. Aus Sicht der Untersuchungsteilnehmer nehmen die Gegebenheiten im Zusammenhang mit den Anforderungen an die Integrationslösung somit eine zentrale Rolle bei der Erklärung des erforderlichen Entwicklungsaufwands ein.

Innerhalb dieses Themenbereichs können verschiedene Schwerpunkte identifiziert werden. Besonders viele der zugeordneten Konstrukte beschreiben die Höhe der *Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung* als Einflussfaktor auf den Entwicklungsaufwand (10 Konstrukte). Diese Konstrukte wurden gleichzeitig mit der höchsten durchschnittlichen Häufigkeit genannt. Mit höheren Anforderungen an die qualitätsbezogenen Merkmale ist gemäß den Erfahrungen der Untersuchungsteilnehmer durchgängig ein höherer Entwicklungsaufwand verbunden. Insbesondere die *Anforderungen an die Sicherheit, Verfügbarkeit und Performanz der Integrationslösung* wurden in diesem Zusammenhang von mehreren Untersuchungsteilnehmern genannt.

Nahezu gleich viele Konstrukte wurden erfasst, die sich auf *die Klarheit der Anforderungen* beziehen (9 Konstrukte), wobei eine geringe Klarheit (bzw. eine Unklarheit) der Anforderungen mit einem höheren Entwicklungsaufwand verbunden wird. Dieser Schwerpunkt kann überraschen, da im Sinne des konzeptionellen Bezugsrahmens dieser Arbeit mit der Anwendungsintegration lediglich vorgegebene Strategien und Geschäftsprozesse unterstützt und bestehende Anwendungssysteme zusammengeführt werden sollen. In den persönlichen Erfahrungen der Untersuchungsteilnehmer spiegelt sich jedoch wider, dass dennoch oftmals unklar bzw. insbesondere unzureichend analysiert und dokumentiert ist, worin das zu lösende Integrationsproblem besteht und auf welche Weise es gelöst werden sollte. Neben der mangelhaften Klarheit der Anforderungsanalyse wurde durch die Teilnehmer oftmals *die Änderungsdynamik der Anforderungen* für einen höheren Entwicklungsaufwand der verglichenen Integrationsvorhaben verantwortlich gemacht (5 Konstrukte).

Ferner bestimmten die Untersuchungsteilnehmer mehrere aufwandsrelevante Unterschiede zwischen den Integrationsvorhaben, die sich auf *die mit der Anwendungsinteg-*

ration verfolgten Perspektive (insbesondere den Integrationsgegenstand) und *die Neuartigkeit der Integrationslösung* beziehen (6 bzw. 5 Konstrukte). Insbesondere mit einer zunehmenden Integration der Prozessabwicklung und einer höheren Neuartigkeit der Integrationslösung verbinden die Untersuchungsteilnehmer einen höheren Entwicklungsaufwand.

Schließlich ist ein höherer Entwicklungsaufwand aus der Sicht mehrerer Untersuchungsteilnehmer mit einem *größeren Umfang und einer größeren Komplexität der Aufgabenstellung aus fachlicher Sicht* zu erklären. Das betreffende Konstrukt wurde in vier Interviews erhoben.

Sämtliche Gruppen der Konstrukte zum Themenbereich der Anforderungen an die Integrationslösung sind mit der jeweiligen Anzahl der ihnen zugeordneten Konstrukte und deren durchschnittlicher Häufigkeit in Tabelle 5-6 dargestellt.

Die einzelnen Konstrukte, die hinter den abgegrenzten Gruppen stehen, werden daraufhin in Tabelle 5-7 dargestellt. Sie vermitteln ein differenziertes Bild des Verständnisses der Untersuchungsteilnehmer von den Einflussfaktoren auf den Entwicklungsaufwand im Zusammenhang mit den Anforderungen an die Integrationslösung. Die beiden polaren Ausprägungen der Konstrukte sind dabei stets in gleicher Weise angeordnet: die Ausprägung, die gemäß der Ansicht der Untersuchungsteilnehmer mit einem geringeren Entwicklungsaufwand verbunden ist, ist in der linken der beiden Spalten aufgeführt; die Ausprägung, die gemäß der Ansicht der Untersuchungsteilnehmer mit einem höheren Entwicklungsaufwand verbunden ist, ist in der rechten der beiden Spalten aufgeführt. Ferner ist die Häufigkeit der einzelnen Konstrukte angegeben.

Die Einschätzungen der Untersuchungsteilnehmer, ob es sich bei den beschriebenen Merkmalen um Charakteristika der Anwendungsintegration handelt und welche relative Bedeutung sie für den Entwicklungsaufwand einnehmen, werden an dieser Stelle im Interesse der Übersichtlichkeit nicht wiedergegeben. Sie werden ausführlich im Anhang dieser Arbeit aufgeführt.

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Der allgemeine Umfang der Aufgabenstellung	2	2,5
Die Anzahl, die Art und die Komplexität der zu unterstützenden Geschäftsprozesse	3	1,3
Die Anzahl und die Komplexität der zu unterstützenden Use Cases	2	1
Die mit der Anwendungsintegration verfolgte Perspektive	6	1
Die Neuartigkeit der Integrationslösung	5	1,4
Die Neuartigkeit der durch die Integrationslösung zu unterstützenden Geschäftsprozesse	2	1
Die Klarheit der Anforderungen	9	1,4
Die Änderungsdynamik der Anforderungen	5	1,6
Der Umgang mit Änderungen der Anforderungen	1	1
Die Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung	10	2,6
Die Anforderungen an die Protokollierung der Interaktionen zwischen den zu integrierenden Anwendungssystemen	2	1
Die Anzahl der Benutzer und Benutzerrollen	2	1
Die Verteilung der Nutzung der Integrationslösung	1	1
Die Notwendigkeit des Abgleichs der Stammdaten der zu integrierenden Anwendungssysteme	3	1
Die Notwendigkeit einer Migration	2	1,5

Tabelle 5-6: Gruppen der Konstrukte zu den Anforderungen an die Integrationslösung

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Der allgemeine Umfang der Aufgabenstellung	Der Integrationsumfang ist gering ("kleiner scope").	Der Integrationsumfang ist groß ("großer scope").	1
	Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht ist gering.	Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht sind groß.	4
Die Anzahl, die Art und die Komplexität der zu unterstützenden Geschäftsprozesse	Die Anzahl der zu unterstützenden Geschäftsprozesse ist gering.	Die Anzahl der zu unterstützenden Geschäftsprozesse ist groß.	2
	Es gilt Geschäftsprozesse einer Art ("Prozessfamilie") zu unterstützen.	Es gilt Geschäftsprozesse vieler unterschiedlicher Arten ("Prozessfamilien") zu unterstützen.	1
	Die Komplexität eines zu unterstützenden Geschäftsprozesses ist gering.	Die Komplexität eines zu unterstützenden Geschäftsprozesses ist groß.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Anzahl und die Komplexität der zu unterstützenden Use Cases	Die Anzahl der zu unterstützenden Use Cases ist gering.	Die Anzahl der zu unterstützenden Use Cases ist groß.	1
	Die Komplexität eines zu unterstützenden Use Cases ist gering.	Die Komplexität eines zu unterstützenden Use Cases ist groß.	1
Die mit der Anwendungsintegration verfolgte Perspektive	Die Anwendungsintegration betrifft ausschließlich technische Aspekte.	Die Anwendungsintegration betrifft technische und fachliche Aspekte.	1
	Die Anwendungssysteme werden integriert, um damit Benutzer zu bedienen. Die Ansprüche an die Anwendungsintegration werden somit durch Menschen gestellt.	Die Anwendungssysteme werden integriert, um damit weitere Softwaresysteme zu bedienen. Die Ansprüche an die Anwendungsintegration werden somit durch Softwaresysteme gestellt.	1
	Die Integrationslösung basiert auf der Daten-Ebene.	Die Integrationslösung umfasst auch Interaktionen mit Benutzern und macht damit insbesondere die Unterstützung von Berechtigungskonzepten und Workflows erforderlich.	1
	Die Anwendungsintegration ist informationsorientiert.	Die Anwendungsintegration ist prozessorientiert.	1
	Die Interaktionen der zu integrierenden Anwendungssysteme erfolgen vollständig automatisiert.	Während der Interaktionen der zu integrierenden Anwendungssysteme sind manuelle Interventionen erforderlich.	1
	Die zu unterstützenden Geschäftsprozesse werden nicht vollständig automatisiert.	Die zu unterstützenden Geschäftsprozesse werden vollständig automatisiert.	1
Die Neuartigkeit der Integrationslösung	Die zu entwickelnde Integrationslösung baut auf einer bereits bestehenden Integrationslösung auf.	Die zu entwickelnde Integrationslösung wird vollständig neu aufgebaut.	3
	Eine bereits bestehende Integrationslösung soll nachgebaut und ersetzt werden.	Bislang ist keine Integrationslösung vorhanden.	1
	Die Integrationslösung basiert auf einer etablierten und verifizierten Architektur.	Die Integrationslösung basiert auf einer nicht ausreichend erprobten Architektur.	1
	Eine bereits zentralisierte Anwendungslandschaft soll durch eine zentrale Lösung integriert werden. Es muss somit eine bestehende IT-Architektur umstrukturiert werden.	Eine verteilte Anwendungslandschaft soll durch die Anwendungsintegration zentralisiert werden. Es muss somit grundsätzlich eine andere IT-Architektur geschaffen werden.	1
	Es ist nicht erforderlich, fachliche Funktionalität neu zu entwickeln.	Es ist erforderlich, fachliche Funktionalität neu zu entwickeln.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Neuartigkeit der durch die Integrationslösung zu unterstützenden Geschäftsprozesse	Die zu unterstützenden Geschäftsprozesse sind etabliert.	Die zu unterstützenden Geschäftsprozesse müssen erst definiert werden.	1
	Ein durch die Integrationslösung unterstützter Geschäftsprozess ist ein Ersatz für einen einheitlichen bestehenden Geschäftsprozess.	Ein durch die Integrationslösung unterstützter Geschäftsprozess ist ein Ersatz für mehrere unterschiedliche Geschäftsprozesse.	1
Die Klarheit der Anforderungen	Der Kunde kennt und versteht das Integrationsproblem.	Der Kunde muss erst eine Integrationslösung sehen und verstehen, um das Integrationsproblem genau zu verstehen.	1
	Es gibt eine klare Aufgabenstellung.	Der Kunde ist sich über seine Ziele unklar. Die Aufgabenstellung entwickelt sich nach und nach.	2
	Die zu unterstützenden Geschäftsprozesse sind klar definiert.	Die zu unterstützenden Geschäftsprozesse sind nicht klar definiert.	1
	Die fachlichen Interdependenzen zwischen den zu integrierenden Anwendungssystemen können genau definiert werden.	Die fachlichen Interdependenzen zwischen den zu integrierenden Anwendungssystemen können nicht definiert werden.	1
	Es erfolgt keine Experimentierung im Laufe des Integrationsvorhabens.	Es erfolgt viel Experimentierung im Laufe des Integrationsvorhabens.	1
	Die Anforderungen sind klar definiert und dokumentiert.	Die Anforderungen sind nicht klar definiert und dokumentiert.	4
	Die Dokumentation der Anforderungen basiert auf formaler Modellierung. Es gibt nur geringen Interpretationsspielraum bezüglich des Analyseergebnisses.	Die Dokumentation der Anforderungen ist lediglich verbalsprachlich. Das Analyseergebnis ist frei interpretierbar.	1
	Die Projektmitarbeiter wissen genau was sie machen müssen.	Die Projektmitarbeiter füllen Lücken in ihrem Verständnis mit eigenen Entscheidungen.	1
	Es gibt eine inhaltliche Interpretation von Qualität. Die Anforderungen an die Qualität der Integrationslösung sind gestaltbar.	Es gibt eine formale Interpretation von Qualität. Die Anforderungen an die Qualität der Integrationslösung sind nicht inhaltlich sondern quantitativ festgelegt.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Änderungsdynamik der Anforderungen	Die Anforderungen sind klar definiert und stabil.	Im Laufe des Integrationsvorhabens treten neue Anforderungen auf.	3
	Es gibt eine klare Aufgabenstellung.	Das Integrationsvorhaben unterliegt einem starken Einfluss von Außen. Es findet in einem dynamischen Umfeld statt.	1
	Das Integrationsvorhaben erfolgt in einem stabilen und bekannten Systemumfeld.	Das Integrationsvorhaben erfolgt in einem dynamischen Systemumfeld. Das Systemumfeld ändert sich im Lauf des Integrationsvorhabens.	1
	Der Projektumfang ändert sich während der Projektlaufzeit nicht.	Der Projektumfang ändert sich während der Projektlaufzeit.	2
	Die Organisation des Kunden ändert sich nicht im Laufe des Integrationsvorhabens.	Die Organisation des Kunden ändert sich im Laufe des Integrationsvorhabens.	1
Der Umgang mit Änderungen der Anforderungen	Es gibt ein etabliertes Change Request System.	Die Verwaltung der Change Requests muss neu erarbeitet und umgesetzt werden.	1
Die Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung (Fortsetzung folgende Seite)	Die qualitätsbezogenen/nicht-funktionalen Anforderungen an die Integrationslösung sind gering.	Die qualitätsbezogenen/nicht-funktionalen Anforderungen an die Integrationslösung sind hoch.	2
	Die Sicherheitsanforderungen an die Integrationslösung sind gering.	Die Sicherheitsanforderungen an die Integrationslösung sind hoch.	4
	Es ist nicht kritisch, dass über die zu integrierenden Anwendungssysteme hinweg Transaktionssicherheit besteht.	Transaktionssicherheit über die zu integrierenden Anwendungssysteme hinweg ist erforderlich.	3
	Die Anforderungen an die Verfügbarkeit der Integrationslösung sind gering.	Die Anforderungen an die Verfügbarkeit der Integrationslösung sind hoch.	4
	Die Anforderungen an die Ausfallsicherheit der Integrationslösung sind gering.	Die Anforderungen an die Ausfallsicherheit der Integrationslösung sind hoch.	2
	Die Anforderungen an die Performanz der Integrationslösung sind gering.	Die Anforderungen an die Performanz der Integrationslösung sind hoch.	4
	Die Anforderungen an die Skalierbarkeit der Integrationslösung sind gering.	Die Anforderungen an die Skalierbarkeit der Integrationslösung sind hoch.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung (Fortsetzung)	Die Anforderungen an die Revisionssicherheit der Integrationslösung sind gering.	Die Anforderungen an die Revisionsicherheit der Integrationslösung sind hoch.	1
	Die Integrationslösung besitzt keine komplexen Fehlerbehandlungsmechanismen. Fehler werden nur protokolliert.	Die Integrationslösung besitzt komplexe Fehlerbehandlungsmechanismen.	2
	Bei der Gestaltung der Integrationslösung werden nur unmittelbare Ziele verfolgt.	Bei der Gestaltung der Integrationslösung werden eine hohe Flexibilität und Wiederverwendbarkeit angestrebt.	3
Die Anforderungen an die Protokollierung der Interaktionen zwischen den zu integrierenden Anwendungssystemen	Bei der Interaktion von Anwendungssystemen besteht kein fachlicher Überwachungsbedarf ("track and trace").	Bei der Interaktion von Anwendungssystemen besteht großer fachlicher Überwachungsbedarf ("track and trace").	1
	Für die Interaktionen der zu integrierenden Anwendungssysteme ist lediglich eine eingeschränkte Kontakthistorie erforderlich.	Für die Interaktionen der zu integrierenden Anwendungssysteme ist eine umfangreiche Kontakthistorie erforderlich.	1
Die Anzahl der Benutzer und Benutzerrollen	Die Anzahl der Benutzer ist gering.	Die Anzahl der Benutzer ist groß.	1
	Die Anzahl der Benutzerrollen ist gering.	Die Anzahl der Benutzerrollen ist groß.	1
Die Verteilung der Nutzung der Integrationslösung	Die Integrationslösung wird zentral an einem Standort genutzt.	Die Integrationslösung wird dezentral an verschiedenen Standorten genutzt.	1
Die Notwendigkeit des Abgleichs der Stammdaten der zu integrierenden Anwendungssysteme	Es ist keine initiale Stammdatensynchronisation erforderlich.	Es ist eine initiale Stammdatensynchronisation erforderlich.	1
	Es ist keine Stammdatensynchronisation zur Laufzeit notwendig.	Es ist eine Stammdatensynchronisation zur Laufzeit erforderlich.	1
	Es gibt einen festen und durchgehenden Satz von Stammdaten.	Die Stammdaten weisen eine hohe Dynamik auf.	1
Die Notwendigkeit einer Migration	Es ist keine Migration von Daten erforderlich.	Es ist eine Migration von Daten erforderlich.	1
	Ein Parallelbetrieb zu migrierender Systeme ist nicht erforderlich.	Ein Parallelbetrieb zu migrierender Systeme ist erforderlich.	2

Tabelle 5-7: Persönliche Konstrukte zu den Anforderungen an die Integrationslösung

5.4.4 Einflussfaktoren im Zusammenhang mit den Charakteristika der zu integrierenden Anwendungssysteme

Dem Themenbereich der Charakteristika der zu integrierenden Anwendungssysteme wurden 12 Gruppen zugeordnet, die insgesamt 35 der persönlichen Konstrukte umfassen. Dabei können wiederum verschiedene inhaltliche Schwerpunkte identifiziert werden.

Die *Anzahl der zu integrierenden Anwendungssysteme* wurde von 6 Untersuchungsteilnehmern als Einflussfaktor auf den Entwicklungsaufwand genannt.

Ähnlich viele Teilnehmer nannten als Ursache für einen höheren Entwicklungsaufwand, wenn die *Schnittstellen der Anwendungssysteme proprietär* sind und somit nicht auf Standards basieren. Das entsprechende Konstrukt wurde in 4 Interviews erhoben.

Mehrere Untersuchungsteilnehmer führten einen höheren Entwicklungsaufwand ferner auf eine geringe *Qualität der Dokumentation und fehlende Ansprechpartner im Hinblick auf die zu integrierenden Anwendungssysteme* und damit den Mangel an geeigneten Informationsquellen zurück (6 Konstrukte). In diesem Zusammenhang wurde insbesondere die *Güte der Dokumentation der Schnittstellen der zu integrierenden Anwendungssysteme* als wichtiger Einflussfaktor genannt.

Nahezu gleich viele Konstrukte wurden erhoben, die eine hohe *Heterogenität der zu integrierenden Anwendungssysteme* als Ursache für einen höheren Entwicklungsaufwand thematisieren (5 Konstrukte). Dabei stellen sowohl Unterschiede in den konzeptionellen als auch in den technischen Charakteristika der Anwendungssysteme wichtige Einflussfaktoren auf den Entwicklungsaufwand dar.

Schließlich wurden vermehrt Merkmale der *Systemumgebung der zu integrierenden Anwendungssysteme* zur Erklärung des jeweiligen Entwicklungsaufwands der verglichenen Integrationsvorhaben angeführt (5 Konstrukte).

Sämtliche Gruppen zu den Charakteristika der zu integrierenden Anwendungssysteme und die erhobenen Konstrukte im Einzelnen sind in den anschließenden Tabellen aufgeführt. Dabei wird die im vorangehenden Kapitel eingeführte Darstellungsform aufgegriffen.⁵⁷⁰

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Anzahl und die Komplexität der zu integrierenden Anwendungssysteme	2	3,5
Die Arten zu integrierender Anwendungssysteme	4	1,0
Die Verfügbarkeit der Schnittstellen der zu integrierenden Anwendungssysteme	3	1,3
Die Eignung der Schnittstellen der zu integrierenden Anwendungssysteme für die Integrationslösung	2	1,5
Die Qualität der Schnittstellen der zu integrierenden Anwendungssysteme	2	1,5
Die Standardbasiertheit der Schnittstellen der zu integrierenden Anwendungssysteme	1	4,0
Die Änderungsdynamik der Schnittstellen der zu integrierenden Anwendungssysteme	1	1,0
Die Notwendigkeit der Anpassung der zu integrierenden Anwendungssysteme	1	2,0
Die Heterogenität der zu integrierenden Anwendungssysteme	5	1,6
Die Systemumgebung der zu integrierenden Anwendungssysteme	5	1,0
Die Qualität der Dokumentation und der Ansprechpartner im Hinblick auf die zu integrierenden Anwendungssysteme	6	1,7
Der Betrieb der zu integrierenden Anwendungssysteme	3	1,0

Tabelle 5-8: Gruppen der Konstrukte zu den Charakteristika der zu integrierenden Anwendungssysteme

⁵⁷⁰ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Anzahl und die Komplexität der zu integrierenden Anwendungssysteme	Die Anzahl der zu integrierenden Anwendungssysteme ist gering.	Die Anzahl der zu integrierenden Anwendungssysteme ist groß.	6
	Die Komplexität eines zu integrierenden Anwendungssystems ist gering.	Die Komplexität eines zu integrierenden Anwendungssystems ist groß.	1
Die Arten zu integrieren der Anwendungssysteme	Im Rahmen der Integrationslösung werden keine Telekommunikationssysteme angebunden.	Im Rahmen der Integrationslösung werden Telekommunikationssysteme angebunden.	1
	Ein zu integrierendes Anwendungssystem ist in räumlicher Hinsicht statisch.	Ein zu integrierendes Anwendungssystem ist in räumlicher Hinsicht mobil (z. B. im Zusammenhang mit Telemetrie).	1
	Eine Interaktion findet zwischen zwei Server-Systemen statt.	Eine Interaktion findet zwischen Server- und Client-Systemen statt.	1
	Alle zu integrierenden Anwendungssysteme sind Standardsoftwaresysteme.	Alle zu integrierenden Anwendungssysteme sind Individualsoftwaresysteme.	1
Die Verfügbarkeit der Schnittstellen der zu integrierenden Anwendungssysteme	Ein zu integrierendes Anwendungssystem hat wohldefinierte Schnittstellen.	Ein zu integrierendes Anwendungssystem hat keine wohldefinierten Schnittstellen.	2
	Die Reife einer Schnittstelle eines Anwendungssystems ist hoch.	Eine Schnittstelle eines Anwendungssystems muss neu entwickelt werden.	1
	Der Kunde hat vollständige Nutzungsmöglichkeiten und -rechte der Schnittstellen eines zu integrierenden Anwendungssystems sowie der zugehörigen Dokumentation.	Der Kunde hat keine Nutzungsmöglichkeiten und -rechte der Schnittstellen eines zu integrierenden Anwendungssystems sowie der zugehörigen Dokumentation.	1
Die Eignung der Schnittstellen der zu integrierenden Anwendungssysteme für die Integrationslösung	Ein zu integrierendes Anwendungssystem ist technisch für die Integration vorbereitet bzw. geeignet.	Ein zu integrierendes Anwendungssystem ist technisch nicht für die Integration vorbereitet bzw. geeignet.	2
	Ein zu integrierendes Anwendungssystem besitzt eine geeignete Schnittstelle für eine erforderliche Online-Verarbeitung.	Eine Online-Verarbeitung muss unter Verwendung einer Batch-Schnittstelle des zu integrierenden Anwendungssystems realisiert werden.	1
Die Qualität der Schnittstellen der zu integrierenden Anwendungssysteme	Die Qualität einer verwendeten Schnittstelle ist hoch.	Die Qualität einer verwendeten Schnittstelle ist gering.	1
	Die Qualität der von einem zu integrierenden Anwendungssystem gelieferten Daten ist hoch.	Die Qualität der von einem zu integrierenden Anwendungssystem gelieferten Daten ist niedrig.	2
Die Standardbasiertheit der Schnittstellen der zu integrierenden Anwendungssysteme	Eine Schnittstelle eines Anwendungssystems basiert auf Standards.	Eine Schnittstelle eines Anwendungssystems ist proprietär.	4

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Änderungsdynamik der Schnittstellen der zu integrierenden Anwendungssysteme	Die verwendeten Schnittstellen sind konstant.	Die verwendeten Schnittstellen unterliegen einer hohen Änderungshäufigkeit.	1
Die Notwendigkeit der Anpassung der zu integrierenden Anwendungssysteme	Ein zu integrierendes Anwendungssystem muss nicht verändert werden.	Ein zu integrierendes Anwendungssystem muss angepasst werden.	2
Die Heterogenität der zu integrierenden Anwendungssysteme	Die zu integrierenden Anwendungssysteme basieren auf den gleichen Technologien.	Die zu integrierenden Anwendungssysteme basieren auf unterschiedlichen Technologien.	2
	Die zu integrierenden Anwendungssysteme sind hinsichtlich ihrer technischen und konzeptionellen Charakteristika einheitlich.	Die zu integrierenden Anwendungssysteme sind hinsichtlich ihrer technischen und konzeptionellen Charakteristika unterschiedlich.	1
	Die zu integrierenden Anwendungssysteme basieren auf einem einheitlichen Datenmodell.	Die zu integrierenden Anwendungssysteme basieren auf unterschiedlichen Datenmodellen.	3
	Es gibt ein homogenes Berechtigungskonzept über die zu integrierenden Anwendungssysteme hinweg.	Es gibt unterschiedliche Berechtigungskonzepte für die zu integrierenden Anwendungssysteme.	1
	Die Schnittstellen der zu integrierenden Anwendungssysteme sind gleichartig.	Die Schnittstellen der zu integrierenden Anwendungssysteme sind unterschiedlich.	1
Die Systemumgebung der zu integrierenden Anwendungssysteme	Ein zu integrierendes Anwendungssystem wird auf einer modernen Plattform betrieben.	Ein zu integrierendes Anwendungssystem wird auf einer unmodernen Plattform betrieben.	1
	Die zu integrierenden Anwendungssysteme sind unabhängig.	Die zu integrierenden Anwendungssysteme sind in eine komplexe gewachsene Anwendungslandschaft eingebunden.	1
	Die Anwendungsintegration betrifft lediglich die Verknüpfung von Servern im Rechenzentrum-Umfeld.	Die Anwendungsintegration erfolgt in einer komplexen Netzwerkwelt. Die Anzahl der zu berücksichtigenden technischen Komponenten ist hoch.	1
	Die zu integrierenden Anwendungssysteme befinden sich in einer einheitlichen Systemumgebung.	Die zu integrierenden Anwendungssysteme befinden sich in unterschiedlichen Systemumgebungen.	1
	Das Umfeld der zu integrierenden Anwendungssysteme ist einheitlich oder muss bei der Integration nicht berücksichtigt werden.	Die zu integrierenden Anwendungssysteme besitzen heterogene Umfelder, die bei der Integration berücksichtigt werden müssen.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Qualität der Dokumentation und der Ansprechpartner im Hinblick auf die zu integrierenden Anwendungssysteme	Die verfügbare Dokumentation ist quantitativ und qualitativ hinreichend.	Es ist keine technische oder fachliche Dokumentation verfügbar.	1
	Die vorgefundene Anwendungslandschaft ist definiert und dokumentiert.	Die vorgefundene Anwendungslandschaft ist nicht definiert und dokumentiert.	1
	Eine Schnittstelle eines Anwendungssystems ist sehr gut dokumentiert.	Eine Schnittstelle eines Anwendungssystems ist nicht dokumentiert.	4
	Es ist möglich, Personen zu finden, welche die zu integrierenden Anwendungssysteme genau kennen.	Es ist nicht möglich, Personen zu finden, welche die zu integrierenden Anwendungssysteme genau kennen.	1
	Es gibt klare vertragliche Regelungen mit dem Hersteller eines zu integrierenden Anwendungssystems.	Ein zu integrierendes Anwendungssystem ist eine alte Individualsoftware. Es gibt keinen klaren Ansprechpartner und kein Know-how für das Anwendungssystem.	1
	Es gibt eine gute Zusammenarbeit mit dem Hersteller eines zu integrierenden Anwendungssystems.	Der Hersteller eines zu integrierenden Anwendungssystems unterstützt das Integrationsvorhaben nicht.	2
Der Betrieb der zu integrierenden Anwendungssysteme	Die zu integrierenden Anwendungssysteme werden selbst betrieben.	Der Betrieb der zu integrierenden Anwendungssysteme ist an externe Dienstleister ausgelagert	1
	Die zu integrierenden Anwendungssysteme sind innerhalb der Organisation.	Die zu integrierenden Anwendungssysteme sind außerhalb der Organisation.	1
	Es herrscht Informationstransparenz im Hinblick auf Betrieb und Nutzung eines zu integrierenden Anwendungssystems.	Es besteht ein Informationsdefizit im Hinblick auf Betrieb und Nutzung eines zu integrierenden Anwendungssystems.	1

Tabelle 5-9: Persönliche Konstrukte zu den Charakteristika der zu integrierenden Anwendungssysteme

5.4.5 Einflussfaktoren im Zusammenhang mit der Gestaltung der Integrationslösung

Dem Themenbereich der Gestaltung der Integrationslösung wurden 19 Gruppen zugeordnet, die insgesamt 49 Konstrukte umfassen. Neben den Anforderungen an die Integrationslösung nimmt dieser Themenbereich damit eine wichtige Rolle im Verständnis der Untersuchungsteilnehmer von den Einflussfaktoren auf den Entwicklungsaufwand ein.

Insbesondere die *Anzahl und Komplexität der Schnittstellen*, anhand derer die Anwendungssysteme und die sonstigen Komponenten der Integrationslösung miteinander verbunden werden, wurden von vielen Untersuchungsteilnehmern als Einflussfaktor genannt. Die entsprechenden Konstrukte wurden in 6 bzw. in 4 Interviews erhoben.

Des Weiteren führten mehrere Untersuchungsteilnehmer einen höheren Entwicklungsaufwand in den verglichenen Integrationsvorhaben auf eine höhere *Komplexität der Datenstrukturen der Integrationslösung* zurück (7 Konstrukte). Dabei bezogen sie sich auf das zugrunde liegende Datenmodell, die Datenbanktabellen bis hin auf die einzelnen Datenelemente.

Ferner wiesen mehrere Teilnehmer auf die Bedeutung der allgemeinen *Komplexität der Integrationsarchitektur* für den Entwicklungsaufwand hin, beispielsweise im Hinblick auf die Anzahl der zu verknüpfenden Komponenten.

Auch *die Anzahl und die Komplexität der zu realisierenden Transformationen* wurden in mehreren Interviews als relevanter Einflussfaktor genannt.

Sämtliche Gruppen zur Gestaltung der Integrationslösung und die einzelnen erhobenen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷¹

⁵⁷¹ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Investition in Entwurf und Architektur der Integrationslösung	3	1
Die Komplexität der Integrationsarchitektur	3	1,7
Die Änderungsdynamik der Integrationsarchitektur	2	1
Die Struktur der Kommunikationsbeziehungen zwischen den zu integrierenden Anwendungssystemen	4	1
Die Anzahl und die Komplexität der Schnittstellen	3	3,7
Die Komplexität der Adapter	1	1
Die Anzahl der Formate der Kommunikation unter den zu integrierenden Anwendungssystemen	4	1
Die Anzahl und die Komplexität der Transformationen	2	2
Die Komplexität der Datenstrukturen der Integrationslösung	7	1
Der Einsatz eines kanonischen Datenmodells für die Durchführung von Transformationen	3	1
Die mit der Integrationslösung zu überwindenden Netzwerkgrenzen	2	1
Die Gestaltung der Kommunikation zwischen den Anwendungssystemen	2	1
Das Ausmaß der Kopplung zwischen den Komponenten der Integrationslösung	3	1,3
Die Realisierung der Steuerung der Interaktionen der zu integrierenden Anwendungssysteme	2	1,5
Die Anzahl und die Komplexität der zu entwickelnden Module der Integrationslösung	2	1
Die Anzahl und die Komplexität der zu realisierenden Funktionen	2	1
Die Gestaltung der Authentifizierung der Benutzer	1	1
Die Berücksichtigung der Fachlichkeit bei der Gestaltung der Integrationslösung	2	1,5
Die Berücksichtigung der Gleichzeitigkeit von Geschäftsprozessen bei der Gestaltung der Integrationslösung	1	1

Tabelle 5-10: Gruppen der Konstrukte zur Gestaltung der Integrationslösung

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Investition in Entwurf und Architektur der Integrationslösung	Die Gruppe der Projektmitarbeiter, die mit der Entwicklung der Architektur betraut sind, ist groß und spielt eine wichtige Rolle im Projekt.	Die Gruppe der Projektmitarbeiter, die mit der Entwicklung der Architektur betraut sind, ist klein und hat lediglich eine beratende Funktion.	1
	Bei der Entwicklung wird eine architektonische Durchgängigkeit sichergestellt.	Es werden lediglich rudimentäre architektonische Vorgaben gemacht.	1
	Fachliches Design, technisches Design und Schnittstellendesign der Integrationslösung werden zentral erstellt und konsequent umgesetzt.	Fachliches Design, technisches Design und Schnittstellendesign der Integrationslösung werden nicht zentral erstellt oder nicht eingehalten.	1
Die Komplexität der Integrationsarchitektur	Die Komplexität der Integrationsarchitektur ist niedrig.	Die Komplexität der Integrationsarchitektur ist groß.	2
	Die Integrationsarchitektur besteht aus wenigen Komponenten.	Die Integrationsarchitektur besteht aus vielen Komponenten.	2
	Die Integrationsarchitektur besteht aus einer Schicht ("one tier architecture").	Die Integrationsarchitektur besteht aus mehreren Schichten ("multi tier architecture").	1
Die Änderungsdynamik der Integrationsarchitektur	Die Integrationsarchitektur ist klar geplant.	Die Integrationsarchitektur ist im Fluss.	1
	Es gibt keine Änderung der Integrationsarchitektur während der Projektlaufzeit.	Die Integrationsarchitektur ändert sich während der Projektlaufzeit.	1
Die Struktur der Kommunikationsbeziehungen zwischen den zu integrierenden Anwendungssystemen	Zwischen den zu integrierenden Anwendungssystemen besteht eine 1-zu-1-Beziehung.	Zwischen den zu integrierenden Anwendungssystemen besteht eine n-zu-m-Beziehung.	1
	Die Kommunikation zwischen den zu integrierenden Anwendungssystemen ist unidirektional.	Die Kommunikation zwischen den zu integrierenden Anwendungssystemen ist bidirektional.	1
	Im Rahmen der Integrationslösung spielen mehrere Anwendungssysteme eine zentrale Rolle.	Im Rahmen der Integrationslösung spielt ein Anwendungssystem eine zentrale Rolle. Die Anbindung anderer Anwendungssysteme orientiert sich an diesem zentralen Anwendungssystem.	1
	Für die Integration der Anwendungssysteme kann ein zentraler Hub eingesetzt werden.	Für die Integration der Anwendungssysteme kann kein zentraler Hub eingesetzt werden.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Anzahl und die Komplexität der Schnittstellen	Die Anzahl der Schnittstellen ist gering.	Die Anzahl der Schnittstellen ist groß.	6
	Die Komplexität einer Schnittstelle ist gering.	Die Komplexität einer Schnittstelle ist groß.	4
	Die Anzahl an Datensätzen, die eine Schnittstelle passieren, ist gering.	Die Anzahl an Datensätzen, die eine Schnittstelle passieren, ist groß.	1
Die Komplexität der Adapter	Die Komplexität eines Adapters ist gering.	Die Komplexität eines Adapters ist groß.	1
Die Anzahl der Formate der Kommunikation unter den zu integrierenden Anwendungssystemen	Im Rahmen der Interaktion von zu integrierenden Anwendungssystemen sind wenige Systemgrenzen zu überwinden und Verarbeitungsschritte durchzuführen.	Im Rahmen der Interaktion von zu integrierenden Anwendungssystemen sind viele Protokoll- und Formatsprünge zu überwinden.	1
	Die Anzahl der Datensatzformate ist gering.	Die Anzahl der Datensatzformate ist groß.	1
	Die Anzahl der Nachrichtenformate ist gering.	Die Anzahl der Nachrichtenformate ist groß.	1
	Die Integration aller Anwendungssysteme basiert auf den gleichen Kommunikationsarten und -formaten.	Die Integration der Anwendungssysteme basiert auf vielen unterschiedlichen Kommunikationsarten und -formaten.	1
Die Anzahl und die Komplexität der Transformationen	Die Anzahl der zu realisierenden Transformationen ("Mappings") ist gering.	Die Anzahl der zu realisierenden Transformationen ("Mappings") ist groß.	3
	Die Komplexität einer zu realisierenden Transformation ("Mapping") ist gering.	Die Komplexität einer zu realisierenden Transformation ("Mapping") ist groß.	1
Die Komplexität der Datenstrukturen der Integrationslösung	Die Komplexität des Datenmodells ist gering.	Die Komplexität des Datenmodells ist groß.	1
	Die Anzahl der Datenbanktabellen ist gering.	Die Anzahl der Datenbanktabellen ist groß.	1
	Die Komplexität einer Datenbanktabelle ist gering.	Die Komplexität einer Datenbanktabelle ist groß.	1
	Die Anzahl der Datenelemente ist gering.	Die Anzahl der Datenelemente ist groß.	1
	Die Komplexität eines Datenelements ist gering.	Die Komplexität eines Datenelements ist groß.	1
	Die Informationsstrukturen sind flach.	Die Informationsstrukturen sind baumartig verschachtelt.	1
	Die Komplexität der Struktur eines Datenobjekts ist gering.	Die Komplexität der Struktur eines Datenobjekts ist groß.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Der Einsatz eines kanonischen Datenmodell für die Durchführung von Transformationen	Die Transformationen ("Mappings") werden anhand eines zentralen ("kanonischen") Datenmodells realisiert.	Die Transformationen ("Mappings") werden individuell zwischen den Anwendungssystemen realisiert.	1
	Es kann ein einziges zentrales ("kanonisches") Datenmodell für die zu integrierenden Anwendungssysteme gefunden werden.	Mehrere zentrale ("kanonische") Datenmodelle, die jeweils für Teilmengen der zu integrierenden Anwendungssysteme gelten, müssen miteinander vereinbart werden.	1
	Es muss kein zentrales ("kanonisches") Datenmodell geschaffen werden.	Es muss ein zentrales ("kanonisches") Datenmodell geschaffen werden.	1
Die mit der Integrationslösung zu überwindenden Netzwerkgrenzen	Die Integration der Anwendungssysteme erfolgt über wenige Netzwerkgrenzen hinweg.	Die Integration der Anwendungssysteme erfolgt über viele Netzwerkgrenzen hinweg.	1
	Die Integration der Anwendungssysteme erfolgt innerhalb einer demilitarisierten Zone/Sicherheitszone.	Die Integration der Anwendungssysteme erfolgt über mehrere demilitarisierten Zonen/Sicherheitszonen hinweg.	1
Die Gestaltung der Kommunikation zwischen den Anwendungssystemen	Die Integration der Anwendungssysteme erfolgt auf der Basis synchroner Mechanismen ("request-reply").	Die Integration der Anwendungssysteme erfolgt auf der Basis asynchroner Mechanismen ("public-subscribe").	1
	Die Kommunikation der zu integrierenden Anwendungssysteme basiert auf Bulk-Lieferungen.	Die Kommunikation der zu integrierenden Anwendungssysteme basiert auf Einzelsatzverarbeitung.	1
Das Ausmaß der Kopplung zwischen den Komponenten der Integrationslösung	Die zu integrierenden Anwendungssysteme werden lose gekoppelt.	Die zu integrierenden Anwendungssysteme werden eng gekoppelt.	2
	Bei der Gestaltung der Integrationslösung wird keine lose Kopplung zwischen den zu integrierenden Anwendungssystemen und den eingesetzten Integrationskomponenten angestrebt. Die Anwendungssysteme "kennen" die verwendete Technologie.	Bei der Gestaltung der Integrationslösung wird eine lose Kopplung zwischen den zu integrierenden Anwendungssystemen und den eingesetzten Integrationskomponenten angestrebt ("Transparenz der Technologie").	1
	Die Integration der Anwendungssysteme erfolgt durch die Nutzung direkter Schnittstellen.	Die Integration der Anwendungssysteme erfolgt durch den Einsatz einer Middleware.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Realisierung der Steuerung der Interaktionen der zu integrierenden Anwendungssysteme	Geschäftsprozesse werden nicht auf der Integrationsplattform abgebildet. Die Steuerung der Interaktionen zwischen den zu integrierenden Anwendungssystemen erfolgt durch die Anwendungssysteme selbst.	Geschäftsprozesse werden auf der Integrationsplattform abgebildet. Die Steuerung der Interaktionen zwischen den zu integrierenden Anwendungssystemen erfolgt über die Integrationsplattform.	1
	Die in der Integrationslösung abzubildenden Geschäftsprozesse und Geschäftsregeln werden fest im Programmcode umgesetzt.	Die in der Integrationslösung abzubildenden Geschäftsprozesse und Geschäftsregeln werden extrahiert und beispielsweise als "business rules" implementiert.	2
Die Anzahl und die Komplexität der zu entwickelnden Module der Integrationslösung	Die Anzahl der zu entwickelnden Module ist gering.	Die Anzahl der zu entwickelnden Module ist groß.	1
	Die Komplexität eines zu entwickelnden Moduls ist gering.	Die Komplexität eines zu entwickelnden Moduls ist groß.	1
Die Anzahl und die Komplexität der zu realisierenden Funktionen	Die Anzahl der zu realisierenden Funktionen ist gering.	Die Anzahl der zu realisierenden Funktionen ist groß.	1
	Die Komplexität einer zu realisierenden Funktion ist gering.	Die Komplexität einer zu realisierenden Funktion ist groß.	1
Die Gestaltung der Authentifizierung der Benutzer	Eine Authentifizierung erfolgt mittels eines geheimen Schlüssels, d. h. ohne eine Passwortsynchronisation.	Eine Authentifizierung erfolgt über Benutzer- und Passwortsynchronisation.	1
Die Berücksichtigung der Fachlichkeit bei der Gestaltung der Integrationslösung	Die Integrationslösung ist an die Geschäftsprozesse und die Betriebsführung der IT des Kunden angepasst.	Die Integrationslösung ist nicht an die Geschäftsprozesse und die Betriebsführung der IT des Kunden angepasst.	2
	Das Zusammenspiel von Technologie und fachlichen Prozessen ist verstanden und kommuniziert.	Das Zusammenspiel von Technologie und fachlichen Prozessen ist nicht verstanden und kommuniziert.	1
Die Berücksichtigung der Gleichzeitigkeit von Geschäftsprozessen bei der Gestaltung der Integrationslösung	Im Falle der Integration auf Datenebene und der Gleichzeitigkeit von Geschäftsprozessen wird der Status vor der Persistierung von Daten geprüft und plausibilisiert.	Im Falle der Integration auf Datenebene und der Gleichzeitigkeit von Geschäftsprozessen wird der Status vor der Persistierung von Daten nicht geprüft und plausibilisiert.	1

Tabelle 5-11: Konstrukte zur Gestaltung der Integrationslösung

5.4.6 Einflussfaktoren im Zusammenhang mit den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen

Dem Themenbereich der eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge wurden 10 Gruppen zugeordnet, die insgesamt 29 Konstrukte umfassen.

In diesem Zusammenhang verwiesen mehrere Untersuchungsteilnehmer darauf, dass der Entwicklungsaufwand zunähme, wenn für die Entwicklung der Integrationslösung *keine Integrationsprodukte verfügbar seien*, die betreffenden Komponenten der Integrationslösung somit individuell entwickelt werden müssten.

Gemäß den Erfahrungen der Teilnehmer stellt sich jedoch nicht bloß die Frage, ob Integrationsprodukte prinzipiell verfügbar sind. Vielmehr wurden mehrere Konstrukte erhoben, die die *Eignung der eingesetzten Integrationsprodukte für die jeweils entwickelte Integrationslösung* als Einflussfaktor auf den Entwicklungsaufwand beschreiben (5 Konstrukte), beispielsweise inwiefern sie die funktions- und qualitätsbezogenen Anforderungen an die Integrationslösung erfüllen. Auch *der Qualität der Integrationsprodukte*, beispielsweise bezogen auf ihren Reifegrad, wird von den Praktikern Bedeutung für den Entwicklungsaufwand beigemessen (4 Konstrukte).

Schließlich führten mehrere Untersuchungsteilnehmer einen höheren Entwicklungsaufwand auf verschiedene *Charakteristika der eingesetzten Technologien* (5 Konstrukte) und das Ausmaß der *Unterstützung durch Entwicklungswerkzeuge* (3 Konstrukte) zurück.

Die Gruppen zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen und die einzelnen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷²

⁵⁷² Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Verfügbarkeit von Integrationsprodukten	4	1,8
Die Schwierigkeit des Verstehens und Verwendens der eingesetzten Integrationsprodukte	2	1
Die Qualität der eingesetzten Integrationsprodukte	4	1,3
Die Änderungsdynamik der eingesetzten Integrationsprodukte	2	1
Die Eignung der eingesetzten Integrationsprodukte für die Integrationslösung	5	1,2
Die Abhängigkeit von den Herstellern der eingesetzten Integrationsprodukte	2	1
Die Unterstützung durch die Hersteller der eingesetzten Integrationsprodukte	1	1
Die eingesetzten Technologien	5	1,4
Die Unterstützung durch Entwicklungswerkzeuge	3	1,3
Die Einhaltung der Standards der Anwendungssysteme	1	1

Tabelle 5-12: Gruppen der Konstrukte zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Verfügbarkeit von Integrationsprodukten	Die Integrationslösung basiert auf standardisierten Integrationsprodukten.	Die Integrationslösung wird vollständig eigenentwickelt.	4
	Die Integrationslösung basiert auf konfigurierbaren Integrationsprodukten.	Die Integrationslösung wird mit Entwicklungswerkzeugen umgesetzt, die keine fertige Lösung produzieren.	1
	Moderne Integrationsprodukte sind verfügbar und geben die Integrationslösung vor.	Es mangelt an Unterstützung der Integrationsaufgaben durch Integrationsprodukte.	1
	Ein Standardadapter steht zur Verfügung.	Ein Adapter muss selber entwickelt werden.	1
Die Schwierigkeit des Verstehens und Verwendens der eingesetzten Integrationsprodukte	Das eingesetzte Integrationsprodukt basiert auf generischen Standards.	Das eingesetzte Integrationsprodukt ist proprietär und sehr speziell.	1
	Das eingesetzte Integrationsprodukt kann sehr leicht installiert/aufgesetzt werden.	Das eingesetzte Integrationsprodukt ist schwer zu installieren/aufzusetzen.	1
Die Qualität der eingesetzten Integrationsprodukte	Der Reifegrad des eingesetzten Integrationsproduktes ist hoch.	Der Reifegrad des eingesetzten Integrationsproduktes ist niedrig.	2
(Fortsetzung folgende Seite)	Das eingesetzte Integrationsprodukt ist etabliert und gut dokumentiert.	Das eingesetzte Integrationsprodukt ist nicht etabliert und wenig dokumentiert.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Qualität der eingesetzten Integrationsprodukte	Das eingesetzte Integrationsprodukt ist stabil und zuverlässig.	Das eingesetzte Integrationsprodukt ist instabil und unberechenbar.	1
(Fortsetzung)	Das eingesetzte Integrationsprodukt kann standardmäßig in der Integrationslösung eingesetzt werden.	Die Standardschnittstellen des eingesetzten Integrationsproduktes sind fehlerhaft.	1
Die Änderungsdynamik der eingesetzten Integrationsprodukte	Es sind keine Updates des eingesetzten Integrationsproduktes erforderlich.	Das eingesetzte Integrationsprodukt weist kurze Updateintervalle auf.	1
	Es ist kein Versionswechsel des eingesetzten Integrationsproduktes während der Projektlaufzeit erforderlich.	Es ist ein Versionswechsel des eingesetzten Integrationsproduktes während der Projektlaufzeit erforderlich.	1
Die Eignung der eingesetzten Integrationsprodukte für die Integrationslösung	Es werden geeignete Integrationsprodukte in der Integrationslösung eingesetzt.	Es werden ungeeignete Integrationsprodukte in der Integrationslösung eingesetzt.	1
	Das eingesetzte Integrationsprodukt erfüllt die gestellten funktions- und qualitätsbezogenen Anforderungen.	Das eingesetzte Integrationsprodukt erfüllt die gestellten funktions- und qualitätsbezogenen Anforderungen nicht.	2
	Es ist keine funktionale Erweiterung des eingesetzten Integrationsproduktes notwendig.	Es werden funktionale Erweiterungen des eingesetzten Integrationsproduktes durch dessen Hersteller erforderlich.	1
	Das eingesetzte Integrationsprodukt bietet tatsächlich die von dem Hersteller genannten Features.	Das eingesetzte Integrationsprodukt bietet nicht die vom Hersteller genannten Features.	1
	Das eingesetzte Integrationsprodukt unterstützt standardmäßig ein erforderliches Nachrichtenformat oder eine anzubindende Schnittstelle.	Das eingesetzte Integrationsprodukt unterstützt ein erforderliches Nachrichtenformat oder eine anzubindende Schnittstelle nicht standardmäßig.	1
Die Abhängigkeit von den Herstellern der eingesetzten Integrationsprodukte	Es gibt ein starkes Lieferantenmanagement bezüglich der eingesetzten Integrationsprodukte.	Es gibt ein schwaches Lieferantenmanagement bezüglich der eingesetzten Integrationsprodukte. Man gerät gegenüber Lieferanten in Abhängigkeiten.	1
	Das eingesetzte Integrationsprodukt stammt aus einer Hand.	Bei dem eingesetzten Integrationsprodukt bestehen Abhängigkeiten von Drittherstellern.	1
Die Unterstützung durch die Hersteller der eingesetzten Integrationsprodukte	Der Hersteller des eingesetzten Integrationsproduktes ist etabliert und hat entsprechende Supportmöglichkeiten.	Der Hersteller des eingesetzten Integrationsproduktes ist klein und hat nur geringe Supportmöglichkeiten.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die eingesetzten Technologien	Die Anzahl der eingesetzten Technologien ist niedrig.	Die Anzahl der eingesetzten Technologien ist groß.	1
	Die eingesetzte Technologie lässt sich durch die Projektmitarbeiter leicht anwenden.	Die eingesetzte Technologie lässt sich durch die Projektmitarbeiter schwer anwenden.	1
	Die eingesetzte Technologie ist etabliert.	Die eingesetzte Technologie ist neuartig.	3
	Die eingesetzte Technologie bietet Unterstützung durch Standardbibliotheken.	Man muss vollständig eigenentwickeln.	1
	Die Integrationslösung basiert auf programmatischer Technologie (d. h. Realisierung mittels Programmiersprache).	Die Integrationslösung basiert auf deklarativer Technologie (d. h. Realisierung mittels Auszeichnungssprachen, wie XML).	1
Die Unterstützung durch Entwicklungswerkzeuge	Es wird ein vorhandenes Entwicklungswerkzeug eingesetzt.	Es wird ein integriertes Entwicklungswerkzeug für die Integrationslösung entwickelt.	1
	Moderne Entwicklungswerkzeuge sind verfügbar.	Moderne Entwicklungswerkzeuge sind nicht verfügbar.	2
	Die Services einer serviceorientierten Architektur werden aufgrund eines generischen Ansatzes durch ein Entwicklungswerkzeug generiert.	Die Services einer serviceorientierten Architektur werden jeweils eigenentwickelt.	1
Die Einhaltung der Standards der Anwendungssysteme	Die Integration eines ERP-Systems erfolgt unter Einhaltung dessen Standards.	Die Integration eines ERP-Systems erfolgt durch Eigenentwicklung bei Abweichung vom Standard des ERP-Systems.	1

Tabelle 5-13: Konstrukte zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen

5.4.7 Einflussfaktoren im Zusammenhang mit dem Testen der Integrationslösung und der Überleitung in die Nutzung

Dem Themenbereich des Testens der Integrationslösung und der Überleitung in die Nutzung wurden 5 Gruppen zugeordnet, die insgesamt 17 der persönlichen Konstrukte umfassen. Damit handelt es sich um den kleinsten der abgegrenzten Themenbereiche.

Im Zusammenhang mit dem Testen haben mehrere Untersuchungsteilnehmer darauf hingewiesen, dass *die Umstände, unter denen die Tests durchgeführt werden*, Einfluss auf den Entwicklungsaufwand besitzen (6 Konstrukte). Beispielsweise ist es gemäß ihrer Erfahrung mit einem höheren Entwicklungsaufwand verbunden, wenn keine Testsysteme der zu integrierenden Anwendungssysteme verfügbar sind und diese somit simuliert werden müssen.

Ferner wurde in mehreren Interviews eine mangelnde *Sorgfalt des Testens* in den verglichenen Integrationsvorhaben für einen letztendlich höheren Entwicklungsaufwand verantwortlich gemacht (4 Konstrukte).

Die Gruppen zum Testen der Integrationslösung und der Überleitung in die Nutzung sowie die einzelnen erhobenen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷³

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Der Umfang des Testens	3	1
Die Sorgfalt des Testens	4	1
Die herrschenden Umstände zur Durchführung des Testens	6	1
Die Verteilung der Zuständigkeiten beim Testen	2	1,5
Das Vorgehen bei der Überleitung in die Nutzung	2	1

Tabelle 5-14: Gruppen der persönlichen Konstrukte zum Testen der Integrationslösung und zur Überleitung in die Nutzung

⁵⁷³ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Der Umfang des Testens	Es werden wenige Teststufen durchgeführt.	Es werden viele Teststufen durchgeführt.	1
	Es werden wenige Regressionstests durchgeführt.	Es werden viele Regressionstests durchgeführt.	1
	Es werden wenige Testfälle durchgeführt.	Es werden viele Testfälle durchgeführt.	1
Die Sorgfalt des Testens	Es gibt eine hohe Testfallüberdeckung bezüglich der Schnittstellen. Es gibt eine lange Testphase.	Es gibt eine geringe Testfallüberdeckung bezüglich der Schnittstellen. Es gibt eine kurze Testphase.	1
	Es wird sehr umfangreich und systematisch getestet.	Es wird nicht sehr umfangreich und systematisch getestet.	1
	Es gibt eine klare zeitliche Einplanung der Integrations-testphase.	Es gibt keine klare zeitliche Einplanung der Integrations-testphase.	1
	Es gibt ein definiertes Testvorgehen.	Es gibt kein definiertes Testvorgehen.	1
Die herrschenden Umstände zur Durchführung des Testens	Eine komplexe Testumgebung muss nicht aufgebaut werden.	Eine komplexe Testumgebung muss aufgebaut werden.	1
	Testsysteme der zu integrierenden Anwendungssysteme sind vorhanden.	Die zu integrierenden Anwendungssysteme müssen für Entwicklung und Test simuliert werden.	1
	Testdaten für die zu integrierenden Anwendungssysteme sind vorhanden.	Testdaten für die zu integrierenden Anwendungssysteme sind nicht vorhanden.	1
	Die Tests sind vollständig automatisiert.	Die Tests sind vollständig manuell.	1
	Erforderliche Korrekturen können unmittelbar vorgenommen werden. Entwicklung und Test durchdringen einander.	Es gibt lange Verzögerungszeiten ("turn-around-time") um Korrekturen vornehmen zu können.	1
	Die eingesetzte Technologie ist Interpreter-basiert. Man kann im laufenden Betrieb Änderungen vornehmen und sofort testen.	Die eingesetzte Technologie ist compiler-basiert. Man muss nach einer Änderung Testdurchläufe starten.	1
Die Verteilung der Zuständigkeiten beim Testen	Die Zuständigkeit für den Test der gesamten Integrationslösung ist gebündelt.	Es gibt getrennte Zuständigkeiten für den Test von Teilen der Integrationslösung.	2
	Es gibt ein internes und vollständig verstandenes Testsystem.	Es gibt verteilte Testsysteme, die getrennt betreut werden.	1
Das Vorgehen bei der Überleitung in die Nutzung	Die Anzahl der durchzuführenden Trainings ist klein.	Die Anzahl der durchzuführenden Trainings ist groß.	1
	Der Auslieferungsprozess ist einfach.	Der Auslieferungsprozess ist komplex.	1

Tabelle 5-15: Konstrukte zum Testen der Integrationslösung und zur Überleitung in die Nutzung

5.4.8 Einflussfaktoren im Zusammenhang mit dem Kunden und den Stakeholdern der Integrationslösung

Dem Themenbereich des Kunden und der Stakeholder des Integrationsvorhabens wurden 7 Gruppen zugeordnet, die insgesamt 34 Konstrukte umfassen.

Das Ausmaß von Erfahrung und Wissen des Kunden wurde in diesem Zusammenhang mehrfach als Einflussfaktor auf den Entwicklungsaufwand beschrieben (6 Konstrukte).

Ebenso häufig wurde *die Gestaltung der Zuständigkeiten auf Kundenseite* mit dem Entwicklungsaufwand der verglichenen Integrationsvorhaben in Verbindung gebracht (6 Konstrukte). Insbesondere ist es gemäß den Erfahrungen der Untersuchungsteilnehmer mit einem geringeren Entwicklungsaufwand verbunden, wenn auf Kundenseite ein zentraler verantwortlicher Ansprechpartner existiert. Das betreffende Konstrukt wurde in 4 Interviews erhoben.

Eine mangelnde *Unterstützung durch den Kunden* wurde von mehreren Teilnehmern für einen höheren Entwicklungsaufwand verantwortlich gemacht (5 Konstrukte).

Ferner haben sie die Erfahrung gemacht, dass *die Einstellung der Stakeholder zum Integrationsvorhaben* durch Interessenkonflikte und abwehrende Haltungen geprägt sein kann, was ebenfalls mit einem höheren Entwicklungsaufwand verbunden sei (4 Konstrukte).

Des Weiteren wurden mehrere Konstrukte erhoben, in denen die *Beziehung zum Kunden* in verschiedener Hinsicht als relevanter Einflussfaktor beschrieben wird (5 Konstrukte).

Schließlich wiesen mehrere Untersuchungsteilnehmer auf die *Homogenität der Stakeholder des Integrationsvorhabens* und in diesem Zusammenhang insbesondere auf die Anzahl der beteiligten Parteien hin, um einen unterschiedlich hohen Entwicklungsaufwand der verglichenen Integrationsvorhaben zu erklären.

Die Gruppen zum Themenbereich des Kunden und der Stakeholder des Integrationsvorhabens sowie die einzelnen erhobenen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷⁴

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Beziehung zum Kunden	5	1
Die Homogenität der Stakeholder des Integrationsvorhabens	4	1,5
Die Einstellung der Stakeholder zum Integrationsvorhaben	4	1
Die Ausprägung der Kommunikation mit dem Kunden	4	1
Das Ausmaß der Unterstützung durch den Kunden	5	1,4
Das Ausmaß von Erfahrung und Wissen des Kunden	6	1
Die Gestaltung der Zuständigkeiten auf Kundenseite	6	1,5

Tabelle 5-16: Gruppen der Konstrukte zum Kunden und den Stakeholdern der Integrationslösung

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Beziehung zum Kunden	Man führt das Integrationsvorhaben als externer Dienstleister für einen Kunden durch.	Man führt das Integrationsvorhaben als interner Dienstleister durch.	1
	Es gibt einen klaren Arbeitsauftrag und eine klare Trennung von Auftraggeber und Auftragnehmer.	Es gibt keinen klaren Arbeitsauftrag und der Kunde ist in die Entwicklung der Integrationslösung involviert.	1
	Man steht seit langem in Beziehung zum Kunden des Integrationsvorhabens.	Das Integrationsvorhaben stellt eine erstmalige oder einmalige Beziehung zum Kunden dar.	1
	Die Integrationslösung wird lediglich entwickelt und bereitgestellt.	Die Projektmitarbeiter leisten Unterstützung ("Support") für die Betriebsseite.	1
	Der Kunde hat Erfahrung darin, Entwicklungsvorhaben gemeinsam mit externen Dienstleistern durchzuführen.	Der Kunde kauft IT-Lösungen als Black-Box ("wie einen PC").	1

⁵⁷⁴ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Homogenität der Stakeholder des Integrationsvorhabens	Die Anzahl der beteiligten Parteien ist gering.	Die Anzahl der beteiligten Parteien ist groß.	3
	Das Integrationsvorhaben betrifft Anwendungssysteme nur einer organisatorischen Einheit.	Das Integrationsvorhaben betrifft Anwendungssysteme mehrerer organisatorischer Einheiten.	1
	Das Integrationsvorhaben betrifft Anwendungssysteme eines Unternehmens.	Das Integrationsvorhaben betrifft Anwendungssysteme mehrerer Unternehmen.	1
	Das Integrationsvorhaben betrifft nur Anwendungssysteme des Kunden.	Das Integrationsvorhaben betrifft Anwendungssysteme des Kunden sowie Anwendungssysteme externer Partner des Kunden.	1
Die Einstellung der Stakeholder zum Integrationsvorhaben	Die Stakeholder des Integrationsvorhabens haben eine gemeinsame Vision.	Die Stakeholder des Integrationsvorhabens haben unterschiedliche Zielsetzungen.	1
	Das Integrationsvorhaben wird durch die Benutzer der zu integrierenden Systeme unterstützt.	Es gibt eine Abwehrhaltung der Benutzer der zu integrierenden Systeme gegen das Integrationsvorhaben.	1
	Die Stakeholder des Integrationsvorhabens haben keine Angst vor damit verbundenen Veränderungen.	Die Stakeholder des Integrationsvorhabens haben Angst vor damit verbundenen Veränderungen.	1
	Die zu entwickelnde Integrationslösung ist vollständig neuartig.	Die zu entwickelnde Integrationslösung konkurriert gegen bestehende Lösungen beim Kunden.	1
Die Ausprägung der Kommunikation mit dem Kunden	Die Anforderungen können unmittelbar mit dem Endkunden abgestimmt werden.	Die Anforderungen des Endkunden werden über Dritte kommuniziert.	1
	Es besteht ausschließlich ein formalisierter Kontakt zwischen den Entwicklern und den Fachabteilungen.	Die Fachabteilungen können die Entwickler direkt kontaktieren.	1
	Es gibt eine starke und formale Einbeziehung der Fachabteilungen.	Die Einbeziehung der Fachabteilungen erfolgt spontan.	1
	Es erfolgt eine fortwährende Kommunikation von Zwischenergebnissen gegenüber dem Kunden.	Dem Kunden werden keine Zwischenergebnisse kommuniziert.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Das Ausmaß der Unterstützung durch den Kunden	Es gibt eine starke Kundenbeteiligung und -mitarbeit.	Es gibt eine sehr geringe Kundenbeteiligung und -mitarbeit.	1
	Der Kunde liefert fachliche Unterstützung bei der Realisierung.	Der Kunde ist nicht in die Realisierung eingebunden.	1
	Der Kunde kann seine Mitwirkungspflicht erfüllen.	Der Kunde kann seine Mitwirkungspflicht nicht erfüllen.	1
	Die Projektmitarbeiter werden durch die Endanwender unterstützt.	Die Projektmitarbeiter werden nicht durch die Endanwender unterstützt.	2
	Die Management Attention für das Integrationsvorhaben ist hoch.	Die Management Attention für das Integrationsvorhaben ist niedrig.	2
Das Ausmaß von Erfahrung und Wissen des Kunden	Der Kunde hat Erfahrung mit der Anwendungsintegration und verfügt über ein klares und einheitliches Begriffssystem zu deren Beschreibung.	Der Kunde hat keine Erfahrung mit der Anwendungsintegration und verfügt über kein klares und einheitliches Begriffssystem zu deren Beschreibung.	1
	Der Kunde hat sich mit der Anwendungsintegration beschäftigt und kennt sich damit aus.	Der Kunde muss bezüglich der Anwendungsintegration "missioniert" werden.	1
	Der Erkenntnisgrad des Managements bezüglich der Probleme von Integrationsvorhaben ist hoch.	Der Erkenntnisgrad des Managements bezüglich der Probleme von Integrationsvorhaben ist gering.	1
	Die Business-Stakeholder besitzen ein hohes IT-Verständnis.	Die Business-Stakeholder besitzen kein IT-Verständnis.	1
	Der Kunde ist mit den zu integrierenden Anwendungssystemen vertraut.	Der Kunde ist mit den zu integrierenden Anwendungssystemen nicht vertraut.	1
	Der Kunde ist fachlich und technisch sehr kompetent.	Der Kunde ist fachlich und technisch eingeschränkt kompetent.	1
Die Gestaltung der Zuständigkeiten auf Kundenseite (Fortsetzung folgende Seite)	Es gibt einen verantwortlichen Ansprechpartner auf Kundenseite.	Es gibt viele verschiedene Ansprechpartner auf Kundenseite.	4
	Man hat unmittelbaren Kontakt zu den Entscheidungsträgern des Kunden.	Die Ansprechpartner beim Kunden haben keine Entscheidungsbefugnis.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Gestaltung der Zuständigkeiten auf Kundenseite (Fortsetzung)	Im Laufe des Integrationsvorhabens ist keine weitere Entscheidung des Kunden notwendig.	Der Kunde hat keine klaren und kompakten Entscheidungsstrukturen. Keiner der Ansprechpartner traut sich eine Entscheidung zu treffen.	1
	Es existiert eine etablierte, mehrheitlich demokratische Entscheidungsstruktur beim Kunden.	Es existiert eine autokratische Entscheidungsstruktur beim Kunden.	1
	Beim Kunden gibt es eine zentrale Leitung der IT.	Beim Kunden erfolgt die Leitung der IT dezentral.	1
	Beim Kunden gibt es eine starke Leitung der IT.	Beim Kunden gibt es eine schwache Leitung der IT.	1

Tabelle 5-17: Persönliche Konstrukte zum Kunde und den Stakeholdern des Integrationsvorhabens

5.4.9 Einflussfaktoren im Zusammenhang mit den Projektmitarbeitern

Dem Themenbereich der Projektmitarbeiter wurden 9 Gruppen zugeordnet, die insgesamt 32 Konstrukte umfassen. In jedem Interview wurde mindestens ein Konstrukt erhoben, das diesem Themenbereich zugeordnet werden kann.

Insbesondere *Erfahrung und Wissen der Projektmitarbeiter* werden in vielen Konstrukten als wichtige Einflussfaktoren auf den Entwicklungsaufwand beschrieben (11 Konstrukte). Neben der allgemeinen Erfahrung wurde in diesem Zusammenhang häufig auf das Fachwissen der Projektmitarbeiter Bezug genommen. Die entsprechenden Konstrukte wurden jeweils in 4 Interviews erhoben.

Auch die *Einstellung der Projektmitarbeiter zum Integrationsvorhaben*, insbesondere deren Bereitschaft, sich auf die neuartigen Charakteristika der Anwendungsintegration einzulassen, ist aus der Sicht der Untersuchungsteilnehmer relevant für den Entwicklungsaufwand (4 Konstrukte).

Gleich viele Konstrukte wurden darüber hinaus erhoben, die eine *Zusammenarbeit mit weiteren Unternehmen bei der Entwicklung der Integrationslösung* mit einem höheren

und nicht etwa mit einem geringeren Entwicklungsaufwand verbinden (4 Konstrukte). Eines der betreffenden Konstrukte wurde gar in 5 Interviews erhoben.

Schließlich wurde in 4 Interviews eine höhere *Anzahl der Projektmitarbeiter* mit einem höheren Entwicklungsaufwand verbunden.

Die Gruppen zu diesem Themenbereich und die einzelnen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷⁵

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Anzahl der Projektmitarbeiter	1	4
Die allgemeine Produktivität der Projektmitarbeiter	1	1
Die Homogenität der Projektmitarbeiter	3	1
Erfahrung und Wissen der Projektmitarbeiter	11	1,7
Die Qualität der Zusammenarbeit unter den Projektmitarbeitern	3	1
Die Kontinuität der Projektmitarbeiter	3	1
Die Einstellung der Projektmitarbeiter zum Integrationsvorhaben	4	1
Die Zusammenarbeit mit weiteren Unternehmen bei der Entwicklung der Integrationslösung	4	2
Die Qualität der allgemeinen Kommunikation unter den Projektbeteiligten	2	1

Tabelle 5-18: Die Gruppen der Konstrukte zu den Projektmitarbeitern

⁵⁷⁵ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Anzahl der Projektmitarbeiter	Es gibt wenige Projektmitarbeiter.	Es gibt viele Projektmitarbeiter.	4
Die allgemeine Produktivität der Projektmitarbeiter	Die Produktivität der Projektmitarbeiter ist hoch.	Die Produktivität der Projektmitarbeiter ist niedrig.	1
Die Homogenität der Projektmitarbeiter	Die Projektmitarbeiter bilden ein homogenes Team mit einem gemeinsamen Verständnis von Begriffen und Konzepten.	Die Projektmitarbeiter bilden lose Teamverbände ohne ein gemeinsames Verständnis von Begriffen und Konzepten.	1
	Die Projektmitarbeiter entstammen dem selben Unternehmen und bilden ein homogenes Team.	Die Projektmitarbeiter entstammen unterschiedlichen Unternehmen und bilden ein heterogenes Team.	1
	Die Projektmitarbeiter formen ein kleines Spezialistenteam.	Die Projektmitarbeiter formen sehr heterogene Teamstrukturen.	1
Erfahrung und Wissen der Projektmitarbeiter (Fortsetzung folgende Seite)	Es ist kein tiefgehendes technisches Spezialwissen erforderlich.	Punktuell ist ein tiefgehendes technisches Spezialwissen erforderlich.	1
	Kompetente Projektmitarbeiter sind verfügbar.	Kompetente Projektmitarbeiter sind nicht verfügbar.	1
	Personen mit Spezialisten-Know-how sind verfügbar.	Personen mit Spezialisten-Know-how sind nicht verfügbar.	1
	Die Projektmitarbeiter haben sehr gutes Fachwissen.	Die Projektmitarbeiter haben kein Fachwissen.	4
	Die Projektmitarbeiter sind mit den zu integrierenden Anwendungssystemen vertraut.	Die Projektmitarbeiter sind nicht mit den zu integrierenden Anwendungssystemen vertraut.	2
	Die Projektmitarbeiter sind erfahren.	Die Projektmitarbeiter sind unerfahren.	4
	Die mit der Analyse betrauten Projektmitarbeiter haben sehr gute Software Engineering Skills ("hard skills").	Die mit der Analyse betrauten Projektmitarbeiter haben keine Software Engineering Skills ("hard skills").	1
	Die Projektmitarbeiter sind mit der eingesetzten Technologie vertraut.	Die Projektmitarbeiter sind mit der eingesetzten Technologie nicht vertraut.	1
	Die Projektmitarbeiter sind Experten im Umgang mit dem eingesetzten Integrationsprodukt.	Die Projektmitarbeiter wissen nicht, welche Features des eingesetzten Integrationsproduktes in der Problemdomäne eingesetzt werden können.	2

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Erfahrung und Wissen der Projektmitarbeiter (Fortsetzung)	Die Anwendungssysteme und Geschäftsprozesse sind den mit dem Testen betrauten Projektmitarbeitern bekannt.	Die Anwendungssysteme und Geschäftsprozesse sind den mit dem Testen betrauten Projektmitarbeitern nicht bekannt.	1
	Die Projektbeteiligten besitzen von Beginn an das erforderliche Wissen.	Die Projektbeteiligten müssen das erforderliche Wissen während der Projektlaufzeit aufbauen.	1
Die Qualität der Zusammenarbeit unter den Projektmitarbeitern	Die Projektmitarbeiter haben eine "Open Source Philosophie" gegenüber anderen Projektmitarbeitern.	Die Projektmitarbeiter gewähren anderen Projektmitarbeitern keinen Einblick in den von ihnen entwickelten Programmcode.	1
	Unter den Projektbeteiligten herrscht Teamgeist.	Unter den Projektbeteiligten herrscht kein Teamgeist.	1
	Die mit der Analyse betrauten Projektmitarbeiter besitzen sehr hohe soziale Kompetenz ("soft skills").	Die mit der Analyse betrauten Projektmitarbeiter sind nicht teamfähig.	1
Die Kontinuität der Projektmitarbeiter	Das Team der Projektmitarbeiter besteht seit längerem.	Das Team der Projektmitarbeiter muss wiederholt aus verschiedenen Quellen zusammengestellt werden.	1
	Die Projektmitarbeiter formen ein eingespieltes Team.	Die Projektmitarbeiter sind nicht miteinander vertraut.	1
	Es gibt dedizierte Mitarbeiter für das Integrationsvorhaben.	Die Mitarbeiter arbeiten parallel auch an projektexternen Aufgaben. Die Ansprechpartner für das Integrationsvorhaben wechseln.	1
Die Einstellung der Projektmitarbeiter zum Integrationsvorhaben	Die Projektmitarbeiter sind motiviert.	Die Projektmitarbeiter sind nicht motiviert.	1
	Die Projektmitarbeiter lösen sich von persönlichen Vorlieben für bestimmte Technologien.	Die Projektmitarbeiter wollen nur in der ihnen vertrauten Art unter Einsatz bestimmter Technologien entwickeln.	1
	Die Projektmitarbeiter sind bereit, sich in neue Systeme/Technologien einzuarbeiten.	Die Projektmitarbeiter sind nicht bereit sich in neue Systeme/Technologien einzuarbeiten.	1
	Die Projektmitarbeiter haben eine gesamtsystemorientierte Perspektive.	Die Projektmitarbeiter haben eine auf Anwendungssysteme fokussierte Perspektive.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Zusammenarbeit mit weiteren Unternehmen bei der Entwicklung der Integrationslösung	Die Realisierung der Integrationslösung wird intern durchgeführt.	Die Realisierung der Integrationslösung wird an externe Dienstleister vergeben.	1
	Die Entwicklung der Integrationslösung wird selbständig durchgeführt.	Die Entwicklung der Integrationslösung erfolgt gemeinsam mit weiteren externen Dienstleistern.	5
	Ein Teil der Verantwortung kann an einen erfahrenen externen Partner abgegeben werden.	Die Gesamtverantwortung liegt in der eigenen Hand.	1
	Die Entwicklung erfolgt in Zusammenarbeit mit einem Unternehmen, mit dem ein partnerschaftliches Verhältnis besteht.	Die Entwicklung erfolgt in Zusammenarbeit mit einem Unternehmen, mit dem ein kommerzielles Verhältnis besteht.	1
Die Qualität der allgemeinen Kommunikation unter den Projektbeteiligten	Die Schlüsselpersonen des Integrationsvorhabens haben eine gute Beziehung zueinander und kommunizieren viel.	Die Schlüsselpersonen des Integrationsvorhabens haben eine schlechte Beziehung zueinander und kommunizieren wenig.	1
	Die Kommunikation zwischen allen Projektbeteiligten ist gewährleistet.	Die einzelnen Projektbeteiligten arbeiten isoliert voneinander jeweils nur an ihren eigenen Arbeitspaketen.	1

Tabelle 5-19: Konstrukte zu den Projektmitarbeitern

5.4.10 Einflussfaktoren im Zusammenhang mit dem Management des Integrationsvorhabens

Dem Themenbereich des Managements des Integrationsvorhabens wurden 4 Gruppen zugeordnet, die insgesamt 22 Konstrukte umfassen.

Insbesondere wurden in mehreren Interviews Aspekte zur Erklärung des mit den verglichenen Integrationsvorhaben verbundenen Entwicklungsaufwands angesprochen, die zusammenfassend als *Qualität des Managements des Integrationsvorhabens* bezeichnet werden können (9 Konstrukte).

Auch die *Gestaltung der Aufbauorganisation des Integrationsvorhabens* wurde unter verschiedenen Gesichtspunkten mit dem Entwicklungsaufwand in Verbindung gebracht

(7 Konstrukte), beispielsweise im Hinblick auf das Ausmaß der Interdependenzen zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter.

Die Gruppen zum Management des Integrationsvorhabens und die einzelnen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷⁶

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Qualität des Managements des Integrationsvorhabens	9	1
Das Ausmaß von vorausschauendem Vorgehen	2	1
Die Gestaltung der Aufbauorganisation des Integrationsvorhabens	7	1,1
Der verfolgte Entwicklungsansatz	4	1

Tabelle 5-20: Gruppen der persönlichen Konstrukte zum Management des Integrationsvorhabens

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Qualität des Managements des Integrationsvorhabens (Fortsetzung folgende Seite)	Es gibt ein starkes Projektmanagement.	Es gibt kein starkes Projektmanagement.	1
	Es gibt ein Projektmanagement mit klarem Scope und klar abgegrenzter Kompetenz.	Es gibt direkte Interventionen des Linienmanagements in das Projekt.	1
	Das Projektmanagement besitzt hohe soziale Kompetenz und fachliches Know-how.	Das Projektmanagement besitzt keine soziale Kompetenz und fachliches Know-how.	1
	Der Projektleiter ist sensibilisiert für Integrationsaspekte.	Der Projektleiter ist nicht sensibilisiert für Integrationsaspekte.	1
	Im Projekt ist ein Risikomanagement implementiert.	Im Projekt ist kein Risikomanagement implementiert.	1
	Es ist von Anfang an ein Changemanagement aufgesetzt.	Es gibt kein Changemanagement.	1
	Das Team der Projektmitarbeiter wird kontrolliert von oben nach unten aufgebaut.	Das Team der Projektmitarbeiter wird zu schnell aufgebaut.	1

⁵⁷⁶ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Qualität des Managements des Integrationsvorhabens	Die Planung des Projektes erfolgt anhand eines Vorgehensmodells.	Die Planung des Projektes erfolgt spontan und durch den Endtermin getrieben.	1
(Fortsetzung)	Das Integrationsvorhaben wird in mehrere Meilensteine eingeteilt und diese Planung wird befolgt.	Das Integrationsvorhaben weist sehr lange Teilprojektphasen auf.	1
Das Ausmaß von vorausschauendem Vorgehen	Das Integrationsvorhaben wurde gut vorbereitet.	Das Integrationsvorhaben wurde unzureichend vorbereitet.	1
	Das Integrationsvorhaben folgt dem Grundsatz "erst denken, dann handeln".	In dem Integrationsvorhaben "sind Denken und Handeln eins".	1
Die Gestaltung der Aufbauorganisation des Integrationsvorhabens	Zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter bestehen keine Interdependenzen.	Zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter bestehen starke Interdependenzen.	2
	Es gibt klare Organisationschnittstellen in der Projektorganisation.	Es gibt eine netzartige Kommunikationsstruktur in der Projektorganisation.	1
	Die Abstimmung und die Kommunikation zwischen Entwicklung und Test werden gefördert und durch dedizierte Projektmitarbeiter ("Integrationsteam") sichergestellt.	Die Abstimmung und die Kommunikation zwischen Entwicklung und Test werden nicht gefördert. Zwischen Entwicklung und Test besteht ein Nadelöhr.	1
	Die Projektorganisation ist teamorientiert.	Die Projektorganisation ist hierarchisch.	1
	Es gibt klar definierte Verantwortlichkeiten hinsichtlich der Integrationsfunktionalität.	Es gibt eine Vermischung der Verantwortlichkeiten hinsichtlich der Integrationsfunktionalität und fachlicher Funktionalität.	1
	Die Verantwortlichkeiten der Projektbeteiligten sind klar definiert. Die organisatorische Gestaltung des Integrationsvorhabens ist projektorientiert, bei Ausrichtung an den zu liefernden Ergebnissen.	Die Verantwortlichkeiten der Projektbeteiligten sind unklar. Das Integrationsvorhaben wird aus der Linienorganisation des Unternehmens heraus durchgeführt. Die Ergebnisse der einzelnen Linien müssen kombiniert werden.	1
	Es gibt ein zentrales Mandat für die Modellierung der zu unterstützenden Geschäftsprozesse.	Das Mandat für die Modellierung der zu unterstützenden Geschäftsprozesse ist auf mehrere Teilprojekte verteilt.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Der verfolgte Entwicklungsansatz	Der Kunde will zu einem bestimmten Termin einen definierten Lieferumfang haben. Es wird ein sequenzieller/dokumentenbasierter Entwicklungsansatz verfolgt.	Der Kunde will zu einem bestimmten Termin einen definierten Lieferumfang haben. Es wird ein evolutionärer Entwicklungsansatz verfolgt.	1
	Der Kunde ist terminorientiert und in der Gestaltung der Funktionalität flexibler. Es wird ein evolutionärer Entwicklungsansatz verfolgt.	Der Kunde ist terminorientiert und in der Gestaltung der Funktionalität flexibler. Es wird ein sequenzieller/dokumentenbasierter Entwicklungsansatz verfolgt.	1
	Unter den Projektbeteiligten herrscht ein Konsens über die verfolgte Entwicklungsmethode.	Unter den Projektbeteiligten herrscht kein Minimalkonsens über die verfolgte Entwicklungsmethode.	1
	Die Releasezyklen der zu integrierenden Anwendungssysteme und die Vorgehensmodelle, auf denen ihre Entwicklung beruht, können im Integrationsvorhaben berücksichtigt werden.	Die zu integrierenden Anwendungssysteme besitzen unterschiedliche Releasezyklen und ihre Entwicklung beruht auf unterschiedlichen Vorgehensmodellen.	1

Tabelle 5-21: Konstrukte zum Management des Integrationsvorhabens

5.4.11 Einflussfaktoren im Zusammenhang mit den Rahmenbedingungen des Integrationsvorhabens

Der letzte der abgegrenzten Themenbereiche bezieht sich auf die Rahmenbedingungen des Integrationsvorhabens. Ihm wurden 10 Gruppen zugeordnet, die insgesamt 37 Konstrukte umfassen.

Insbesondere die *räumliche Verteilung der Projektbeteiligten* wurde in vielen Interviews als signifikanter Einflussfaktor auf den Entwicklungsaufwand bezeichnet. Auf diesen Aspekt bezieht sich auch das insgesamt am häufigsten erhobene Konstrukt (es wurde in 8 Interviews genannt).

Auch die *Internationalität eines Integrationsvorhabens* und in diesem Zusammenhang ein unterschiedlicher *Sprach- und Kulturraum der Projektbeteiligten* wurden von meh-

reren Untersuchungsteilnehmern für einen höheren Entwicklungsaufwand verantwortlich gemacht.

Vielfach wurde ein unterschiedlicher Entwicklungsaufwand der verglichenen Integrationsvorhaben zudem anhand der *bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben* erklärt (8 Konstrukte). Beispielsweise wird es mit einem höheren Entwicklungsaufwand verbunden, wenn Vorgaben bezüglich der einzusetzenden Integrationsprodukte oder der zu verwendenden Schnittstellen existieren, die für die Lösung des Integrationsproblems prinzipiell nicht geeignet oder notwendig erscheinen.

Schließlich wiesen mehrere Untersuchungsteilnehmer auf die Bedeutung der *Klarheit der Zielsetzung des Integrationsvorhabens* (5 Konstrukte) oder die *Abhängigkeit des Integrationsvorhabens von anderen Projekten* (5 Konstrukte) für den Entwicklungsaufwand hin.

Die Gruppen zu den Rahmenbedingungen des Integrationsvorhabens und die einzelnen Konstrukte sind in den anschließenden Tabellen aufgeführt.⁵⁷⁷

Gruppe	Anzahl der Konstrukte	Ø Häufigkeit der Konstrukte
Die Dauer des Integrationsvorhabens	2	2
Die Neuartigkeit des Integrationsvorhabens	1	1
Die Klarheit der Zielsetzung des Integrationsvorhabens	5	1
Die Bedeutung des Integrationsvorhabens für den Kunden	3	1,3
Das Ausmaß der Internationalität des Integrationsvorhabens	4	2,8
Die räumliche Verteilung der Projektbeteiligten	2	5
Die verfügbare Infrastruktur zur Durchführung des Integrationsvorhabens	4	1,3
Die Abhängigkeit des Integrationsvorhabens von anderen Projekten	5	1
Die bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben	8	1,6
Das verfügbare Budget / die zulässige Entwicklungsdauer	3	1

Tabelle 5-22: Gruppen der Konstrukte zu den Rahmenbedingungen des Integrationsvorhabens

⁵⁷⁷ Zur gewählten Darstellungsform vgl. Kapitel 5.4.3.

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die Dauer des Integrationsvorhabens	Das Integrationsvorhaben hat eine kurze Laufzeit.	Das Integrationsvorhaben hat eine lange Laufzeit.	2
	Das Integrationsvorhaben umfasst wenige Releases.	Das Integrationsvorhaben umfasst viele Releases.	2
Die Neuartigkeit des Integrationsvorhabens	Das Integrationsvorhaben stellt die Fortführung eines vorangehenden Projektes dar.	Das Integrationsvorhaben ist vollständig neu.	1
Die Klarheit der Zielsetzung des Integrationsvorhabens	Das Integrationsvorhaben ist auf die Bedürfnisse des Einzelfalls ausgerichtet.	Das Integrationsvorhaben soll als Vorbild/Vorreiter für ein verändertes Vorgehen in der IT dienen.	1
	Das Integrationsvorhaben verfolgt klar abgegrenzte Zielvorgaben.	Das Integrationsvorhaben stellt einen ersten Schritt auf einem langen Weg dar.	1
	Die Entwicklung einer umfassenden Integrationsarchitektur wurde explizit als Infrastrukturprojekt aufgesetzt.	Das Vorhaben, eine umfassende Integrationsarchitektur zu entwickeln, ist historisch gewachsen.	1
	Neben der Anwendungsintegration sind wenige begleitenden Maßnahmen durchzuführen.	Neben der Anwendungsintegration sind viele begleitenden Maßnahmen durchzuführen.	1
	Mit der Anwendungsintegration werden Kostenvorteile auf hoher Budgetebene angestrebt.	Das Integrationsvorhaben ist charakterisiert durch ein permanentes Streben nach Kostenvorteilen auch im Kleinen.	1
Die Bedeutung des Integrationsvorhabens für den Kunden	Die Anwendungsintegration hat lediglich interne Auswirkungen für die beteiligten Unternehmen.	Die Anwendungsintegration hat eine Außenwirkung im Hinblick auf den Endkunden.	2
	Ein Misserfolg des Integrationsvorhabens ist für den Kunden nicht geschäftsgefährdend.	Das Integrationsvorhaben ist erfolgskritisch für den Kunden.	1
	Die zu integrierenden Anwendungssysteme sind nicht geschäftskritisch.	Die zu integrierenden Anwendungssysteme sind geschäftskritisch.	1
Das Ausmaß der Internationalität des Integrationsvorhabens	Das Integrationsvorhaben ist national.	Das Integrationsvorhaben ist international.	4
	Die Dokumentation der Integrationslösung erfolgt in einer Sprache.	Die Dokumentation der Integrationslösung erfolgt in mehreren Sprachen.	1
	Die Projektbeteiligten sind mit der Projektsprache sehr vertraut.	Die Projektbeteiligten sind mit der Projektsprache nicht vertraut.	2
	Die Projektbeteiligten entstammen einem gemeinsamen Sprach- und Kulturraum.	Die Projektbeteiligten entstammen unterschiedlichen Sprach- und Kulturräumen.	4
Die räumliche Verteilung der Projektbeteiligten	Die Projektbeteiligten befinden sich an einem Standort.	Die Projektbeteiligten sind räumlich getrennt.	8
	Die Projektbeteiligten befinden sich in derselben Zeitzone.	Die Projektbeteiligten befinden sich in unterschiedlichen Zeitzeonen.	2

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die verfügbare Infrastruktur zur Durchführung des Integrationsvorhabens	Es gibt angemessene getrennte Umgebungen für Entwicklung, Test und Abnahme.	Die Entwicklung der Integrationslösung erfolgt unmittelbar in der Produktivumgebung.	2
	Die Produktivumgebung kann in Entwicklungs- und Testumgebungen nachgebildet werden.	Die Produktivumgebung kann nicht nachgebildet werden. Entwicklungs- und Testumgebungen entsprechen nicht der Produktivumgebung.	1
	Die Integrationslösung muss lediglich in der Produktivumgebung eingeführt ("deployed") werden.	Die Integrationslösung muss in Entwicklungs-, Test- und Produktivumgebungen eingeführt ("deployed") werden.	1
	Die räumliche und technische Ausstattung der Projektmitarbeiter ist optimal.	Es ist keine angemessene räumliche und technische Infrastruktur für die Projektmitarbeiter vorhanden.	1
Die Abhängigkeit des Integrationsvorhabens von anderen Projekten	Bei dem Integrationsvorhaben handelt es sich um ein eigenständiges Projekt.	Das Integrationsvorhaben ist ein Teilaspekt eines größeren, umfassenden Projektes.	1
	Es gibt keine Parallelprojekte von denen man abhängig ist.	Es gibt Parallelprojekte von denen man abhängig ist.	1
	Es ist keine Abstimmung mit anderen Projekten erforderlich.	Es ist eine Abstimmung mit anderen Projekten erforderlich.	1
	Man kann die Releaseplanung für das Integrationsvorhaben eigenständig vornehmen.	Das Integrationsvorhaben muss in eine bestehende Releaseplanung eingebettet werden.	1
	Das Integrationsvorhaben ist ein Teilaspekt eines Projektes mit kurzer Gesamtdauer bzw. geringem Gesamtvolumen.	Das Integrationsvorhaben ist ein Teilaspekt eines Projektes mit langer Gesamtdauer bzw. großem Gesamtvolumen.	1
Die bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben (Fortsetzung folgende Seite)	Eine beim Kunden bereits eingesetzte Integrationsplattform ist für die Integrationslösung vorgegeben.	Man hat Wahlfreiheit bezüglich eines neu einzuführenden Integrationsproduktes.	1
	Man hat Wahlfreiheit bezüglich der einzusetzenden Integrationsprodukte und der verwendeten Schnittstellen.	Es gibt Vorgaben bezüglich der einzusetzenden Integrationsprodukte oder Schnittstellen, die für die Lösung des Integrationsproblems nicht geeignet oder notwendig sind.	3
	Es gibt klare Vorgaben des Kunden für die Entwicklung ("starke Governance" beim Kunden).	Die Entwicklung wird nicht durch den Kunden kontrolliert ("keine starke Governance" beim Kunden).	2
	Es gibt keine bestimmten Anforderungen des Kunden, wie eine Software zu entwickeln ist.	Es gibt bestimmte Anforderungen des Kunden, wie eine Software zu entwickeln ist.	1

Gruppe	Persönliche Konstrukte		Häufigkeit
	Weniger Aufwand wenn...	Höherer Aufwand wenn...	
Die bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben (Fortsetzung)	Der Kunde mischt sich nicht in die Entwicklung ein.	Der Kunde mischt sich in die Entwicklung ein.	2
	Das Integrationsvorhaben bezweckt die einmalige Verbindung der zu integrierenden Anwendungssysteme.	Der Lebenszyklus der zu integrierenden Anwendungssysteme muss bei der Entwicklung der Integrationslösung berücksichtigt werden. Systematische Veränderungen der zu integrierenden Anwendungssysteme (insbesondere der Schnittstellen) müssen unterstützt werden	1
	Die Integrationslösung wird aufgrund eigener Konzepte maßgeschneidert.	Die Integrationslösung basiert auf einem standardisierten Paradigma, dessen Umsetzung einen Overhead bedeutet.	2
	Datenschutzrechtliche Bestimmungen sind von geringer Bedeutung.	Datenschutzrechtliche Bestimmungen sind von großer Bedeutung.	1
Das verfügbare Budget / die zulässige Entwicklungsdauer	Das Budget des Kunden ist knapp. Der Kunde hat beschränkte finanzielle Mittel.	Das Budget des Kunden ist ausreichend. Der Kunde hat große finanzielle Mittel.	1
	Man kann bedächtig entwickeln.	Es herrscht großer Zeitdruck.	1
	Es gibt zwar einen vorgegebenen Plan für den Projektverlauf aber man kann Abweichungen von den Meilensteinen einkalkulieren.	Es gibt vorgegebene Release-termine, die nicht verändert werden können.	1

Tabelle 5-23: Persönliche Konstrukte zu den Rahmenbedingungen des Integrationsvorhabens

5.5 Die Untersuchungsergebnisse im Kontext bisheriger Erkenntnisse

Im Weiteren werden die vorgestellten Untersuchungsergebnisse mit den bisherigen Erkenntnissen zu den Einflussfaktoren auf den Entwicklungsaufwand von Softwareentwicklungsvorhaben verglichen. In dieser Arbeit wurden diesbezüglich stellvertretend verschiedene Untersuchungen und Verfahren aus unterschiedlichen Teildisziplinen der Softwareentwicklung vorgestellt. Die Studie von Walston und Felix, das COCOMO II-Verfahren, das COCOTS-Verfahren, das COSOSIMO-Verfahren und die Untersuchung zum integrationsorientierten Reengineering von Altsystemen werden in diesem Kapitel erneut aufgegriffen, um inhaltliche Gemeinsamkeiten mit den erhobenen Konstrukten zu ermitteln.⁵⁷⁸

Darüber hinaus sollen auch die im konzeptionellen Bezugsrahmen vorgestellten Beiträge zu den Erfolgsfaktoren und Risiken für Integrationsvorhaben in den Vergleich mit einbezogen werden.⁵⁷⁹ Es erscheint interessant, inwiefern inhaltliche Gemeinsamkeiten zwischen ihnen und den in den Konstrukten beschriebenen Einflussfaktoren auf den Entwicklungsaufwand bestehen, denn es wurde bereits in der Erläuterung des Praxisproblems darauf hingewiesen, dass der Entwicklungsaufwand und dessen zuverlässige Schätzung von wesentlicher Bedeutung für den Erfolg eines Integrationsvorhabens sind.⁵⁸⁰ Faktoren, die zu einem ungeplant hohen Entwicklungsaufwand führen, können somit als Risiken für den Erfolg eines Integrationsvorhabens wahrgenommen werden.

In Tabelle 5-24 werden die Gruppen der erhobenen Konstrukte den ausgewählten Beiträgen gegenübergestellt. Inhaltliche Gemeinsamkeiten sind durch ein Häkchen markiert. Dieses soll zum Ausdruck bringen, dass mindestens zwischen einem Konstrukt der betreffenden Gruppe und einem Faktor der betreffenden Untersuchung oder des betreffenden Verfahrens inhaltliche Gemeinsamkeiten bestehen. Der Fall, dass sämtliche Konstrukte einer Gruppe eine Entsprechung in den Faktoren eines der Beiträge finden, ist dagegen nie eingetreten.⁵⁸¹

⁵⁷⁸ Vgl. Kapitel 2.4.3.

⁵⁷⁹ Vgl. Kapitel 2.3.3.

⁵⁸⁰ Vgl. Kapitel 1.1.1.

⁵⁸¹ Die gewählte Darstellungsform orientiert sich an Whyte, Bytheway /Information System Success/ 89f.

Gruppe der erhobenen Konstrukte	Walston, Felix ⁵⁸²	COCOMO II ⁵⁸³	COCOTS ⁵⁸⁴	COSOSIMO ⁵⁸⁵	Integrationsorientiertes Reengineering ⁵⁸⁶	Erfolgsfaktoren ⁵⁸⁷	Risikofaktoren ⁵⁸⁸
Gruppen der Konstrukte zu den Anforderungen an die Integrationslösung							
Der allgemeine Umfang der Aufgabenstellung							
Die Anzahl, die Art und die Komplexität der zu unterstützen- den Geschäftsprozesse							
Die Anzahl und die Komplexität der zu unterstützenden Use Cases				✓			
Die mit der Anwendungsintegration verfolgte Perspektive							
Die Neuartigkeit der Integrationslösung		✓		✓			
Die Neuartigkeit der durch die Integrationslösung zu unter- stützenden Geschäftsprozesse							
Die Klarheit der Anforderungen	✓	✓				✓	✓
Die Änderungsdynamik der Anforderungen	✓	✓		✓			✓
Der Umgang mit Änderungen der Anforderungen							
Die Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung	✓	✓	✓				
Die Anforderungen an die Protokollierung der Interaktionen zwischen den zu integrierenden Anwendungssystemen							
Die Anzahl der Benutzer und Benutzerrollen							
Die Verteilung der Nutzung der Integrationslösung							
Die Notwendigkeit des Abgleichs der Stammdaten der zu integrierenden Anwendungssysteme							
Die Notwendigkeit einer Migration							
Der allgemeine Umfang der Aufgabenstellung							
Gruppen der Konstrukte zu den Charakteristika der zu integrierenden Anwendungssysteme							
Die Anzahl und die Komplexität der zu integrierenden An- wendungssysteme				✓		✓	
Die Arten zu integrierender Anwendungssysteme							
Die Verfügbarkeit der Schnittstellen der zu integrierenden Anwendungssysteme							✓

⁵⁸² Vgl. Walston, Felix /Programming Measurement and Estimation/ 55-72, bzw. Kapitel 2.4.3.2 dieser Arbeit.

⁵⁸³ Vgl. Boehm u. a. /COCOMO II/, bzw. Kapitel 2.4.3.2 dieser Arbeit.

⁵⁸⁴ Vgl. Abts, Boehm, Clark /COCOTS/, bzw. Kapitel 2.4.3.3 dieser Arbeit.

⁵⁸⁵ Es wurde sowohl das Early-Design Model als auch das Post-Architecture Model des COSOSIMO-Verfahrens in den Vergleich miteinbezogen. Vgl. Lane /System-of-Systems/, bzw. Kapitel 2.4.3.4 dieser Arbeit.

⁵⁸⁶ Vgl. Jung /Integrationsorientiertes Reengineering/, bzw. Kapitel 2.4.3.5 dieser Arbeit.

⁵⁸⁷ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/, bzw. Kapitel 2.3.3.1 dieser Arbeit.

⁵⁸⁸ Vgl. Vgl. Lam /Enterprise Integration/, bzw. Kapitel 2.3.3.2 dieser Arbeit.

Gruppe der erhobenen Konstrukte	Walston, Felix ⁵⁸²	COCOMO II ⁵⁸³	COCOTS ⁵⁸⁴	COSOSIMO ⁵⁸⁵	Integrationsorientiertes Reengineering ⁵⁸⁶	Erfolgsfaktoren ⁵⁸⁷	Risikofaktoren ⁵⁸⁸
Die Eignung der Schnittstellen der zu integrierenden Anwendungssysteme für die Integrationslösung				✓			✓
Die Qualität der Schnittstellen der zu integrierenden Anwendungssysteme					✓	✓	
Die Standardbasiertheit der Schnittstellen der zu integrierenden Anwendungssysteme							
Die Änderungsdynamik der Schnittstellen der zu integrierenden Anwendungssysteme				✓			
Die Notwendigkeit der Anpassung der zu integrierenden Anwendungssysteme			✓	✓	✓		✓
Die Heterogenität der zu integrierenden Anwendungssysteme				✓		✓	
Die Systemumgebung der zu integrierenden Anwendungssysteme						✓	
Die Qualität der Dokumentation und der Ansprechpartner im Hinblick auf die zu integrierenden Anwendungssysteme					✓	✓	
Der Betrieb der zu integrierenden Anwendungssysteme							
Gruppen der Konstrukte zur Gestaltung der Integrationslösung							
Die Investition in Entwurf und Architektur der Integrationslösung		✓	✓	✓		✓	✓
Die Komplexität der Integrationsarchitektur			✓		✓	✓	
Die Änderungsdynamik der Integrationsarchitektur	✓			✓			
Die Struktur der Kommunikationsbeziehungen zwischen den zu integrierenden Anwendungssystemen							
Die Anzahl und die Komplexität der Schnittstellen			✓	✓			
Die Komplexität der Adapter							
Die Anzahl der Formate der Kommunikation unter den zu integrierenden Anwendungssystemen							
Die Anzahl und die Komplexität der Transformationen							
Die Komplexität der Datenstrukturen der Integrationslösung					✓		
Der Einsatz eines kanonischen Datenmodells für die Durchführung von Transformationen							✓
Die mit der Integrationslösung zu überwindenden Netzwerk-grenzen							
Die Gestaltung der Kommunikation zwischen den Anwendungssystemen							
Das Ausmaß der Kopplung zwischen den Komponenten der Integrationslösung				✓		✓	
Die Realisierung der Steuerung der Interaktionen der zu integrierenden Anwendungssysteme						✓	
Die Anzahl und die Komplexität der zu entwickelnden Module der Integrationslösung							
Die Anzahl und die Komplexität der zu realisierenden Funktionen		✓		✓			

Gruppe der erhobenen Konstrukte	Walston, Felix ⁵⁸²	COCOMO II ⁵⁸³	COCOTS ⁵⁸⁴	COSOSIMO ⁵⁸⁵	Integrationsorientiertes Reengineering ⁵⁸⁶	Erfolgsfaktoren ⁵⁸⁷	Risikofaktoren ⁵⁸⁸
Die Gestaltung der Authentifizierung der Benutzer							
Die Berücksichtigung der Fachlichkeit bei der Gestaltung der Integrationslösung						✓	
Die Berücksichtigung der Gleichzeitigkeit von Geschäftsprozessen bei der Gestaltung der Integrationslösung							✓
Gruppen der Konstrukte zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen							
Die Verfügbarkeit von Integrationsprodukten					✓	✓	
Die Schwierigkeit des Verstehens und Verwendens der eingesetzten Integrationsprodukte			✓				
Die Qualität der eingesetzten Integrationsprodukte			✓			✓	✓
Die Änderungsdynamik der eingesetzten Integrationsprodukte							
Die Eignung der eingesetzten Integrationsprodukte für die Integrationslösung							✓
Die Abhängigkeit von den Herstellern der eingesetzten Integrationsprodukte							✓
Die Unterstützung durch die Hersteller der eingesetzten Integrationsprodukte			✓				
Die eingesetzten Technologien						✓	✓
Die Unterstützung durch Entwicklungswerkzeuge		✓	✓	✓	✓		
Die Einhaltung der Standards der Anwendungssysteme							
Gruppen der Konstrukte zum Testen der Integrationslösung und zur Überleitung in die Nutzung							
Der Umfang des Testens							
Die Sorgfalt des Testens	✓						
Die herrschenden Umstände zur Durchführung des Testens					✓		
Die Verteilung der Zuständigkeiten beim Testen							
Das Vorgehen bei der Überleitung in die Nutzung							
Gruppen der Konstrukte zu den Kunden und den Stakeholdern							
Die Beziehung zum Kunden							✓
Die Homogenität der Stakeholder des Integrationsvorhabens				✓			
Die Einstellung der Stakeholder zum Integrationsvorhaben		✓					
Die Ausprägung der Kommunikation mit dem Kunden							✓
Das Ausmaß der Unterstützung durch den Kunden	✓	✓				✓	
Das Ausmaß von Erfahrung und Wissen des Kunden							
Die Gestaltung der Zuständigkeiten auf Kundenseite						✓	

Gruppe der erhobenen Konstrukte	Walston, Felix ⁵⁸⁹	COCOMO II ⁵⁹⁰	COCOTS ⁵⁹¹	COSOSIMO ⁵⁹²	Integrationsorientiertes Reengineering ⁵⁹³	Erfolgsfaktoren ⁵⁹⁴	Risikofaktoren ⁵⁹⁵
Gruppen der Konstrukte zu den Projektmitarbeitern							
Die Anzahl der Projektmitarbeiter	✓						
Die allgemeine Produktivität der Projektmitarbeiter							
Die Homogenität der Projektmitarbeiter		✓					
Erfahrung und Wissen der Projektmitarbeiter	✓	✓	✓	✓	✓	✓	✓
Die Qualität der Zusammenarbeit unter den Projektmitarbeitern		✓		✓			
Die Kontinuität der Projektmitarbeiter		✓	✓	✓			
Die Einstellung der Projektmitarbeiter zum Integrationsvorhaben							
Die Zusammenarbeit mit weiteren Unternehmen bei der Entwicklung der Integrationslösung							
Die Qualität der allgemeinen Kommunikation unter den Projektbeteiligten		✓		✓			
Gruppen der Konstrukte zum Management des Integrationsvorhabens							
Die Qualität des Managements des Integrationsvorhabens		✓		✓		✓	✓
Das Ausmaß von vorausschauendem Vorgehen						✓	
Die Gestaltung der Aufbauorganisation des Integrationsvorhabens						✓	✓
Der verfolgte Entwicklungsansatz				✓			

⁵⁸⁹ Vgl. Walston, Felix /Programming Measurement and Estimation/ 55-72, bzw. Kapitel 2.4.3.2 dieser Arbeit.

⁵⁹⁰ Vgl. Boehm u. a. /COCOMO II/, bzw. Kapitel 2.4.3.2 dieser Arbeit.

⁵⁹¹ Vgl. Abts, Boehm, Clark /COCOTS/, bzw. Kapitel 2.4.3.3 dieser Arbeit.

⁵⁹² Es wurde sowohl das Early-Design Model als auch das Post-Architecture Model des COSOSIMO-Verfahrens in den Vergleich miteinbezogen. Vgl. Lane /System-of-Systems/, bzw. Kapitel 2.4.3.4 dieser Arbeit.

⁵⁹³ Vgl. Jung /Integrationsorientiertes Reengineering/, bzw. Kapitel 2.4.3.5 dieser Arbeit.

⁵⁹⁴ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/, bzw. Kapitel 2.3.3.1 dieser Arbeit.

⁵⁹⁵ Vgl. Vgl. Lam /Enterprise Integration/, bzw. Kapitel 2.3.3.2 dieser Arbeit.

Gruppe der erhobenen Konstrukte	Walston, Felix ⁵⁹⁶	COCOMO II ⁵⁹⁷	COCOTS ⁵⁹⁸	COSOSIMO ⁵⁹⁹	Integrationsorientiertes Reengineering ⁶⁰⁰	Erfolgsfaktoren ⁶⁰¹	Risikofaktoren ⁶⁰²
Gruppen der Konstrukte zu den Rahmenbedingungen des Integrationsvorhabens							
Die Dauer des Integrationsvorhabens	✓						
Die Neuartigkeit des Integrationsvorhabens							
Die Klarheit der Zielsetzung des Integrationsvorhabens						✓	✓
Die Bedeutung des Integrationsvorhabens für den Kunden				✓			
Das Ausmaß der Internationalität des Integrationsvorhabens							
Die räumliche Verteilung der Projektbeteiligten		✓					
Die verfügbare Infrastruktur zur Durchführung des Integrationsvorhabens	✓	✓		✓			
Die Abhängigkeit des Integrationsvorhabens von anderen Projekten							
Die bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben							
Das verfügbare Budget / die zulässige Entwicklungsdauer		✓		✓			

Tabelle 5-24: Vergleich der Gruppen der erhobenen Konstrukte mit bestehenden Erkenntnissen zum Entwicklungsaufwand von Softwareentwicklungsvorhaben

Die Gegenüberstellung macht deutlich, dass die ausgewählten bisherigen Erkenntnisse zu den Einflussfaktoren auf den Entwicklungsaufwand der Vielfalt der erhobenen Konstrukte kaum gerecht werden. Dies gilt insbesondere für die Erkenntnisse zur Neuentwicklung von Anwendungssystemen (vertreten durch die Studie von Walston und Felix sowie das COCOMO II-Verfahren). Beispielsweise finden die von den Untersuchungsteilnehmern genannten Einflussfaktoren im Zusammenhang mit den Charakteristika der zu integrierenden Anwendungssysteme, mit der Gestaltung der Integrationslösung oder

⁵⁹⁶ Vgl. Walston, Felix /Programming Measurement and Estimation/ 55-72, bzw. Kapitel 2.4.3.2 dieser Arbeit.

⁵⁹⁷ Vgl. Boehm u. a. /COCOMO II/, bzw. Kapitel 2.4.3.2 dieser Arbeit.

⁵⁹⁸ Vgl. Abts, Boehm, Clark /COCOTS/, bzw. Kapitel 2.4.3.3 dieser Arbeit.

⁵⁹⁹ Es wurde sowohl das Early-Design Model als auch das Post-Architecture Model des COSOSIMO-Verfahrens in den Vergleich miteinbezogen. Vgl. Lane /System-of-Systems/, bzw. Kapitel 2.4.3.4 dieser Arbeit.

⁶⁰⁰ Vgl. Jung /Integrationsorientiertes Reengineering/, bzw. Kapitel 2.4.3.5 dieser Arbeit.

⁶⁰¹ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/, bzw. Kapitel 2.3.3.1 dieser Arbeit.

⁶⁰² Vgl. Vgl. Lam /Enterprise Integration/, bzw. Kapitel 2.3.3.2 dieser Arbeit.

mit den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen nur wenige Entsprechungen. Aber auch die in den Beiträgen zum Einsatz von Standardsoftwarekomponenten, zur Entwicklung eines SoS oder zum integrationsorientierten Reengineering von Altsystemen behandelten Faktoren decken lediglich Ausschnitte der erhobenen Konstrukte ab.

Auffällig ist in diesem Zusammenhang, dass keiner der Untersuchungsteilnehmer die Anzahl der zu entwickelnden Programmzeilen oder logischen Anweisungen beim Vergleich der Integrationsvorhaben als relevanten Unterschied angesprochen hat. Dies kann eine Folge der rückblickenden Auseinandersetzung mit den Projekten sein, bei der Details des Programmcodes „übersehen“ wurden. Womöglich waren den Untersuchungsteilnehmern diese Informationen zu den betrachteten Integrationsvorhaben nicht bekannt. Wahrscheinlich bringt die fehlende Erwähnung dieser Merkmale des Programmcodes jedoch zum Ausdruck, dass sie von den Teilnehmern nicht als zentrale Aspekte für die Erklärung des Entwicklungsaufwands von Integrationsvorhaben angesehen werden.

Des Weiteren wird die Vermutung bestätigt, dass die Untersuchungsergebnisse inhaltliche Gemeinsamkeiten mit den Beiträgen zu den Erfolgsfaktoren und Risiken für Integrationsvorhaben aufweisen. Vor allem im Zusammenhang mit den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen sowie mit dem Management der Integrationsvorhaben erscheint das Ausmaß an Übereinstimmung größer als im Hinblick auf die bisherigen Erkenntnisse zu den Einflussfaktoren auf den Entwicklungsaufwand. Insgesamt gilt jedoch auch hier, dass die Vielfalt der erhobenen Konstrukte lediglich zu einem geringen Teil abgebildet wird.

Aus dem durchgeführten Vergleich kann der folgende Schluss gezogen werden: Die dargestellten Untersuchungen und Verfahren zur Aufwandschätzung, die den bisherigen Stand der Erkenntnis repräsentieren, werden den Erfahrungen und Ansichten erfahrener Praktiker zu den Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration erwartungsgemäß nicht gerecht.

Dass darüber hinaus keine weiterführenden Erkenntnisse aus dem Vergleich gewonnen werden können, ist auf verschiedene Einschränkungen zurückzuführen.

Zunächst konnten die erhobenen Konstrukte lediglich einem kleinen Ausschnitt der bisherigen Erkenntnisse zu den Einflussfaktoren auf den Entwicklungsaufwand gegenübergestellt werden.⁶⁰³ Ferner bleibt die relative Bedeutung der Einflussfaktoren für den Entwicklungsaufwand unberücksichtigt. Es wäre daher möglich, dass die festgestellten inhaltlichen Gemeinsamkeiten gerade diejenigen Faktoren mit dem größten Einfluss auf den Entwicklungsaufwand von Integrationsvorhaben betreffen, wogegen die anderen von den Untersuchungsteilnehmern genannten Einflussfaktoren von geringerer Bedeutung sind. Dieser Schluss erscheint jedoch, auch vor dem Hintergrund der Kommentierung der Konstrukte durch die Teilnehmer, sehr unwahrscheinlich.⁶⁰⁴ Des Weiteren können die für den Vergleich ausgewählten Beiträge infolge der ihnen zugrunde liegenden Untersuchungsgegenstände, Begriffsverständnisse und Annahmen nur eingeschränkt auf die vorliegende Untersuchung übertragen werden oder aber sie beziehen sich lediglich auf Teilaspekte der Anwendungsintegration im Sinne dieser Arbeit. Ferner wurde bereits kritisiert, dass die eingangs postulierten Anforderungen an die Entwicklung valider und zuverlässiger Maße in vielen Fällen nicht erfüllt sind und die intersubjektive Nachvollziehbarkeit der Untersuchungen, die Verifizierung und Repräsentativität der Ergebnisse oftmals nicht in angemessener Weise sichergestellt ist.⁶⁰⁵ Es ist daher kaum zu beurteilen, ob festgestellte inhaltliche Gemeinsamkeiten auf besonders relevante Einflussfaktoren und fehlende inhaltliche Gemeinsamkeiten auf Besonderheiten der Anwendungsintegration oder Mängel der bisherigen Erkenntnisse hindeuten.

⁶⁰³ Dabei sind allein durch die geringere Anzahl der in den ausgewählten Beiträgen behandelten Faktoren klare Grenzen für die Suche nach Entsprechungen für die erhobenen Konstrukte gesetzt.

⁶⁰⁴ Die Einschätzungen der Untersuchungsteilnehmer sprechen dafür, dass mit der Untersuchung spezifische Einflussfaktoren identifiziert werden konnten. Sie beurteilen 55 % der Konstrukte als Beschreibungen von Charakteristika von Integrationsvorhaben. Ferner messen die Untersuchungsteilnehmer den identifizierten Einflussfaktoren überwiegend eine hohe und mittlere Bedeutung für den Entwicklungsaufwand zu. Schließlich wiesen in diesem Zusammenhang mehrere Teilnehmer darauf hin, dass solche Einflussfaktoren, die auch für die klassische Softwareentwicklung relevant seien, ihrer Erfahrung nach in Integrationsvorhaben eine andere Bedeutung für den Entwicklungsaufwand einnehmen können. Wenn die Einflussfaktoren als Knotenpunkte in einem komplexen Netzwerk aus Ursachen und Wirkungen verstanden werden, könnten demnach in Integrationsvorhaben spezifische Pfade zwischen den Einflussfaktoren und dem Entwicklungsaufwand bestehen.

⁶⁰⁵ Vgl. Kapitel 3.5.

6 Beurteilung der Untersuchung und Schlussfolgerungen

In diesem Kapitel werden die durchgeführte Untersuchung und die gewonnenen Ergebnisse dem Erkenntnisziel der Arbeit gegenübergestellt. Dazu wird im ersten Schritt die durchgeführte Untersuchung beurteilt. Im zweiten Schritt werden die wissenschaftlichen Implikationen der Untersuchungsergebnisse diskutiert und Anregungen für weiterführende Untersuchungen gegeben. Im dritten Schritt wird schließlich die Relevanz der Untersuchungsergebnisse für die praktische Aufwandschätzung von Integrationsvorhaben aufgezeigt.

6.1 Beurteilung der Untersuchung

Gemäß Steinke erfordert die Beurteilung von Untersuchungsergebnissen im Falle qualitativer Forschungsansätze die Anwendung spezifischer Kriterien: die Angemessenheit der Untersuchung zum Erreichen des Erkenntnisziels (indication of the research process), die intersubjektive Nachvollziehbarkeit (inter-subject comprehensibility), die Konsistenz der Untersuchungsergebnisse (coherence), die Verdeutlichung der Einschränkungen der Erkenntnisse (limitation) und die Relevanz der Untersuchungsergebnisse (relevance).⁶⁰⁶ Im Folgenden sollen die ersten Kriterien auf die vorliegende Untersuchung angewandt werden. Auf die Relevanz der Untersuchungsergebnisse für Wissenschaft und Praxis wird darüber hinaus in den anschließenden Kapiteln eingegangen.

6.1.1 Angemessenheit der Untersuchung

Mit der vorliegenden Untersuchung sollten das Verständnis und insbesondere die Interpretationsmuster erfahrener Praktiker bezüglich der Einflussfaktoren auf den mit Integrationsvorhaben verbundenen Entwicklungsaufwand erhoben werden. Aufgrund der bisherigen Erkenntnisse zum Entwicklungsaufwand der Softwareentwicklung muss von einer großen Anzahl, auf komplizierte Weise miteinander in Beziehung stehender Einflussfaktoren ausgegangen werden. Es war anzunehmen, dass sich diese Komplexität auch im Verständnis und den Interpretationsmustern erfahrener Praktiker widerspiegeln würde. Vor dem Hintergrund der wenig fortgeschrittenen wissenschaftlichen Auseinandersetzung mit dem Entwicklungsaufwand der Anwendungsintegration war es daher das Anliegen der Untersuchung, möglichst vielfältige persönliche Erfahrungen zu erfassen.

⁶⁰⁶ Zu den von Steinke empfohlenen Gütekriterien qualitativer Forschung vgl. im Weiteren Steinke /Qualitative Research/.

Mit der vorliegenden Untersuchung wurde daher ein qualitativer und idiographischer Forschungsansatz verfolgt.⁶⁰⁷

Im Sinne der Theorie der persönlichen Konstrukte spiegelt sich das Verständnis einer Person von ihrer Umwelt in dem System ihrer persönlichen Konstrukte wider. Dieses Konstruktsystem erklärt auch die von der Person oftmals unbewusst angewandten Interpretationsmuster. Es galt daher eine Untersuchungsmethode zu wählen, die es ermöglicht, die persönlichen Konstrukte erfahrener Praktiker zu aufwandsrelevanten Sachverhalten explizit zu machen. Angesichts ihrer erläuterten Stärken empfiehlt sich diesbezüglich die angewandte Methode zur Erhebung persönlicher Konstrukte. Verschiedene Autoren haben bereits von ihrem erfolgreichen Einsatz in der Disziplin der Wirtschaftsinformatik berichtet. Die Ausgestaltung dieser Untersuchungsmethode, die Festlegung der Anforderungen an die Auswahl der Auskunftspersonen und der durch sie zu erörternden Integrationsvorhaben ebenso wie die Präzisierung des Bezugsrahmens für die Vergleiche orientieren sich konsequent an den zuvor definierten Untersuchungszielen. Mit der Spezifizierung der Untersuchungsmethode und der sorgfältigen Vorbereitung wurden geeignete Voraussetzungen für die Durchführung der Interviews und die Erhebung reichhaltiger und zuverlässiger Informationen geschaffen. Auch bei der anschließenden inhaltlichen Analyse der erhobenen Konstrukte wurde stets darauf geachtet, die Empfehlungen verschiedener Wissenschaftler zur Gewährleistung der Validität der Untersuchungsergebnisse aufzugreifen. Die Untersuchung erscheint damit insgesamt als geeignet, das Erkenntnisziel der Arbeit zu erreichen.⁶⁰⁸

6.1.2 Intersubjektive Nachvollziehbarkeit der Untersuchung

Gemäß Steinke kann im Falle qualitativer Forschungsansätze, im Unterschied zu quantitativen Untersuchungen, keine intersubjektive *Nachprüfbarkeit* (Verifizierbarkeit) erreicht werden, da eine identische Wiederholung der Untersuchungen nicht möglich ist. Vielmehr fordert Steinke, die intersubjektive *Nachvollziehbarkeit* sicherzustellen.⁶⁰⁹

⁶⁰⁷ Dagegen erschien eine quantitative Messung der Verteilung bestimmter Ansichten unter den Praktikern, um darüber im Sinne eines normativen Forschungsansatzes auf Gesetzmäßigkeiten ihrer Interpretationsmuster zu schließen, dem Stand der Erkenntnis gegenüber nicht als angemessen.

⁶⁰⁸ Vgl. Steinke /Qualitative Research/ 188f.

⁶⁰⁹ Zum Kriterium der intersubjektiven Nachvollziehbarkeit vgl. im Weiteren Steinke /Qualitative Research/ 186f.

Grundlegend für die Nachvollziehbarkeit einer Untersuchung ist ihre präzise Dokumentation. Diesbezüglich wird es als notwendig angesehen, das vorausgehende Verständnis der Untersuchenden vom Untersuchungsgegenstand, die angewandte Untersuchungsmethode, die durchgeführte Nachbereitung und Analyse der erhobenen Informationen, die verwendeten Informationsquellen, die im Rahmen der Untersuchung angefallenen Probleme und die gewählten Lösungswege und schließlich die erfüllten Gütekriterien differenziert zu erläutern. Diesen Forderungen wird in der vorliegenden Arbeit mit der umfangreichen Erläuterung des konzeptionellen Bezugsrahmens, der detaillierten Beschreibung der Untersuchung und der gewonnenen Ergebnisse sowie mit der an dieser Stelle durchgeführten Beurteilung der Untersuchung Rechnung getragen. Auf diese Weise wird es dem Leser ermöglicht, die Untersuchung nachzuvollziehen und über diese Ausführungen hinaus eigene Kriterien zu ihrer Beurteilung anzuwenden.⁶¹⁰

Weitere Grundlagen der intersubjektiven Nachvollziehbarkeit werden mit der Berücksichtigung der Urteile mehrerer Personen und dem Explizitmachen der getroffenen Interpretationen genannt. Sie erscheinen insbesondere im Zusammenhang mit der Analyse der erhobenen Informationen von Bedeutung.⁶¹¹ Der ersten Anforderung wird durch die Plausibilisierung der durchgeführten inhaltlichen Analyse und insbesondere durch die Validierung sämtlicher Analyseergebnisse durch die Untersuchungsteilnehmer entsprochen. Indem im Zuge der Darstellung der Untersuchungsergebnisse nicht nur Themenbereiche und Gruppen, sondern auch die einzelnen Konstrukte aufgeführt wurden, ist es dem Leser darüber hinaus möglich, die getroffenen Interpretationen nachzuvollziehen oder die erhobenen Konstrukte selbständig anhand anderer Gesichtspunkte auszuwerten.

Schließlich wird die intersubjektive Nachvollziehbarkeit der Untersuchung durch die Anwendung einer definierten und verbreiteten Untersuchungsmethode unterstützt. Spezifika der in der vorliegenden Untersuchung gewählten Vorgehensweise wurden stets aufgezeigt und begründet. Die vorliegende Arbeit kann somit vor dem Hintergrund anderer Beiträge zur Repertory Grid Technique und zur Erhebung persönlicher Konstrukte beurteilt werden.⁶¹²

⁶¹⁰ Vgl. Steinke /Qualitative Research/ 187.

⁶¹¹ Vgl. Steinke /Qualitative Research/ 187.

⁶¹² Zur Anwendung vereinheitlichter Untersuchungsmethoden zur Schaffung intersubjektiver Nachvollziehbarkeit vgl. Steinke /Qualitative Research/ 188.

6.1.3 Konsistenz der Untersuchungsergebnisse

Die Untersuchungsteilnehmer haben zusammenfassend betrachtet ein sehr komplexes aber in sich konsistentes Verständnis der für den Entwicklungsaufwand relevanten Merkmale von Integrationsvorhaben aufgezeigt. Es konnten viele inhaltliche Gemeinsamkeiten zwischen den erhobenen Konstrukten festgestellt werden. Die Teilnehmer haben dabei die beschriebenen Aspekte auf einheitliche Weise in Beziehung zu dem erforderlichen Entwicklungsaufwand gesetzt. Insbesondere trat nie der Fall ein, dass ein Konstrukt durch mehrere Untersuchungsteilnehmer erhoben wurde, dann jedoch unterschiedliche Konstruktpole mit einem höheren Entwicklungsaufwand in Verbindung gebracht worden wären.

Wie erläutert bilden die persönlichen Konstrukte eines Individuums ein komplexes hierarchisches System. Die in der vorliegenden Untersuchung erhobenen Konstrukte bilden diesbezüglich lediglich einzelne Ausschnitte aus den Konstruktsystemen der Untersuchungsteilnehmer ab. Dies führt dazu, dass die erhobenen Konstrukte die Merkmale von Integrationsvorhaben mit einem unterschiedlichen Detailgrad beschreiben. Während sich einige Konstrukte auf grundlegende Zusammenhänge beziehen, wie beispielsweise „Der Integrationsumfang ist gering (‚kleiner scope‘) – Der Integrationsumfang ist groß (‚großer scope‘)“, scheinen andere Konstrukte konkretere Einzelheiten zu betreffen, wie beispielsweise „Die Anzahl der zu unterstützenden Geschäftsprozesse ist gering – Die Anzahl der zu unterstützenden Geschäftsprozesse ist groß“.

Diese Unterschiede sollten jedoch nicht als Inkonsistenzen der Untersuchungsergebnisse, sondern als Ausdruck der Überordnungs- und Unterordnungsverhältnisse in den Konstruktsystemen angesehen werden. Für weiterführende Untersuchungen könnte es sich sogar als hilfreich erweisen, dass das Verständnis der Untersuchungsteilnehmer von den Einflussfaktoren auf den Entwicklungsaufwand auf unterschiedlichen Abstraktionsebenen erfasst wurde. Die Untersuchungsergebnisse vermitteln damit einen vielschichtigen Eindruck von den Interpretationsmustern erfahrener Praktiker, was sowohl für die wissenschaftliche Auseinandersetzung mit den realen Phänomenen als auch für die Entwicklung geeigneter Methoden für die Aufwandschätzung von Nutzen sein kann.

6.1.4 Einschränkungen der Untersuchungsergebnisse

Die dargestellten Untersuchungsergebnisse unterliegen verschiedenen Einschränkungen, die aus den Charakteristika der Auskunftspersonen und der betrachteten Integrationsvorhaben sowie etwaigen Verzerrungseffekten der Untersuchungsmethode resultieren.

Aufgrund der Anzahl der Auskunftspersonen, der Vielfalt ihrer persönlichen Erfahrungshintergründe und der Verschiedenartigkeit der von ihnen betrachteten Integrationsvorhaben wird davon ausgegangen, dass mit der vorliegenden Untersuchung ein breites Spektrum der zweckbezogenen persönlichen Interpretationsmuster erfahrener Praktiker erfasst werden konnte.

Gleichwohl können die dargestellten Untersuchungsergebnisse nicht den Anspruch auf Vollständigkeit erheben. Zum einen wurde bereits erläutert, dass durch die Methode zur Erhebung persönlicher Konstrukte lediglich ein Ausschnitt des komplexen und dynamischen Konstruktsystems eines Individuums abgebildet werden kann. Es muss daher davon ausgegangen werden, dass mit der Durchführung weiterer Vergleiche ein größerer Ausschnitt der Konstruktsysteme der Untersuchungsteilnehmer hätte erfasst werden können. Dem ist entgegenzustellen, dass die Teilnehmer die Auswahl der Integrationsvorhaben und die Vergleiche nach eigenem Ermessen durchführten, sie also selbständig darüber entschieden, welcher Ausschnitt ihrer Konstruktsysteme erhoben wurde. Dass es sich dabei aus Sicht der Teilnehmer um relevante Erkenntnisse handelt, wird anhand des großen Anteils der Konstrukte deutlich, denen sie eine große oder mittlere Bedeutung für den Entwicklungsaufwand zumaßen.

Des Weiteren kann nicht ausgeschlossen werden, dass durch zusätzliche Auskunftspersonen neuartige Konstrukte erhoben werden könnten. Die Trendanalyse der inhaltlichen Gemeinsamkeiten der Konstrukte deutet jedoch darauf hin, dass wesentliche Merkmale der Integrationsvorhaben erfasst wurden.

Auch nehmen die Untersuchungsergebnisse keine Repräsentativität für sich in Anspruch. Es wurde darauf hingewiesen, dass der Umfang der Population der Untersuchungssubjekte nicht bekannt ist. Ferner ist nicht mit Gewissheit abzusehen, welche Merkmale der Untersuchungssubjekte für deren Verständnis von den Einflussfaktoren des Entwicklungsaufwands relevant sind oder wie diese in der Population verteilt sind. Im Sinne des theoretischen Samplings konnte es daher nicht das Anliegen der Untersu-

chung sein, gemeingültige Aussagen über die Interpretationsmuster aller Praktiker im Kontext der Anwendungsintegration zu ermöglichen. Vielmehr sollten mit dem idiographischen Forschungsansatz möglichst vielfältige persönliche Erfahrungswerte der Untersuchungsteilnehmer beschrieben werden.

In diesem Zusammenhang wird es als Stärke der angewandten Methode zur Erhebung der persönlichen Konstrukte angesehen, dass die Erfahrungswerte der Untersuchungsteilnehmer unmittelbar in ihrem eigenen Wortlaut erfasst werden können. Trotz der sorgfältigen Vorbereitung und der angestrebten Vorsicht des Interviewers kann eine Beeinflussung der Teilnehmer bei der Durchführung der Vergleiche und der Bezeichnung der Konstruktpole jedoch nicht gänzlich ausgeschlossen werden. Wie erläutert, entwickelten die Teilnehmer in vielen Fällen erst im Dialog mit dem Interviewer ein präzises Bewusstsein für ihr implizites Verständnis von den maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand. Die Beeinflussung des Auskunftsverhaltens kann als prinzipielle Problematik persönlicher Befragungssituationen angesehen werden. Ihr steht jedoch auch der Vorteil gegenüber, dass im Unterschied zu schriftlichen Erhebungen motivierende oder Unklarheiten beseitigende Erläuterungen gegeben werden konnten.⁶¹³ In diesem Sinne betonten mehrere Untersuchungsteilnehmer, dass sie im Falle einer schriftlichen Erhebung keine vergleichbar intensive Auseinandersetzung mit dem Untersuchungsgegenstand erreicht hätten.

Unabhängig vom Einfluss des Interviewers könnten die Auskünfte der Untersuchungsteilnehmer zudem durch die rückblickende Auseinandersetzung mit den Integrationsvorhaben verzerrt worden sein. In diesem Zusammenhang äußerte einer der Teilnehmer im Anschluss an das Interview, dass ihm aus heutiger Sicht zwischenmenschliche Herausforderungen der Integrationsvorhaben bewusster seien und dringlicher erschienen als technische Spezifika der Projekte. Weitere Teilnehmer betonten im Verlauf der Interviews, dass sie insbesondere solche Aspekte der Integrationsvorhaben erörtern würden, die in Zusammenhang mit der Koordination und Kommunikation unter den Projektbeteiligten stehen. Technische Probleme seien dagegen rückblickend immer zu lösen gewesen und besäßen daher aus ihrer Sicht eine geringere Bedeutung für den Entwicklungsaufwand.

⁶¹³ Vgl. Müller /Prüf- und Testprozesse/ 117.

Die rückwirkende Auseinandersetzung mit den Integrationsvorhaben könnte somit zu einem überproportionalen Anteil von Konstrukten führen, welche die Arbeitsteilung und das Auftraggeber-Auftragnehmer-Verhältnis in den Integrationsvorhaben betreffen. Es fällt jedoch schwer, einen solchen verzerrenden Effekt der Methode anhand der Untersuchungsergebnisse nachzuweisen. Generell kann nicht ohne weiteres von einer Verfälschung der Untersuchungsergebnisse ausgegangen werden - schließlich zeigen auch frühere Untersuchungen die zentrale Bedeutung zwischenmenschlicher Aspekte für den Aufwand der Entwicklung und Bereitstellung eines Softwaresystems.⁶¹⁴

Schließlich unterliegt die Absicht, über Interviews die Erfahrungen und Einschätzungen von Auskunftspersonen zu erheben und dadurch Rückschlüsse auf die Realität zu ziehen, prinzipiellen Einschränkungen. Die menschliche Interpretation der Realität ebenso wie die Erinnerung an vergangene Ereignisse sind durch vielfältige Verzerrungen und Unvollkommenheit gekennzeichnet.⁶¹⁵ Es muss daher betont werden, dass aufgrund der Untersuchung nicht reale Phänomene nachgewiesen werden können, sondern ausschließlich ein Einblick in die Interpretationsmuster erfahrener Praktiker gewonnen wurde. Inwiefern dadurch Ansatzpunkte für weiterführende Untersuchungen geschaffen wurden, wird im anschließenden Kapitel erläutert.

6.2 Wissenschaftliche Relevanz der Untersuchungsergebnisse

Mit der Untersuchung sollten neue Erkenntnisse zu den Einflussfaktoren auf den Entwicklungsaufwand von Integrationsvorhaben gewonnen werden. Auf diese Weise sollte insbesondere eine Basis für die Generierung empirischer Hypothesen geschaffen werden. Im Hinblick auf diese Zielsetzung wird im Weiteren auf die wissenschaftlichen Implikationen der Untersuchungsergebnisse eingegangen.

6.2.1 Erkenntnisbeitrag der Untersuchung

Die vorliegende Untersuchung liefert erstmals systematisch gewonnene empirische Erkenntnisse zur Beantwortung der Frage nach den maßgeblichen Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration.

⁶¹⁴ Vgl. beispielsweise Stevenson /Software Engineering Productivity/ 541f.; Gilb /Software Engineering Management/ 256f.

⁶¹⁵ Vgl. Rugg u. a. /Laddering/ 216f.

In Interviews mit 22 erfahrenen Praktikern wurden 310 Konstrukte erhoben, welche die persönlichen Erfahrungen der Untersuchungsteilnehmer und die von ihnen angewandten Interpretationsmuster bei der Auseinandersetzung mit Integrationsvorhaben beschreiben. Gemäß den eigenen Auskünften der Teilnehmer ist es dabei gelungen, auch implizites Wissen zu den Einflussfaktoren auf den Entwicklungsaufwand zu erfassen.

Die durchgeführte Trendanalyse der inhaltlichen Gemeinsamkeiten der Konstrukte deutet darauf hin, dass die für den Entwicklungsaufwand als maßgebend angesehenen Merkmale von Integrationsvorhaben zumindest in großen Teilen erfasst werden konnten. Des Weiteren weisen die Einschätzungen der Untersuchungsteilnehmer darauf hin, dass überwiegend Einflussfaktoren mit einer hohen oder mittleren Bedeutung für den Entwicklungsaufwand dokumentiert wurden.

Der durchgeführte inhaltliche Vergleich der erhobenen Konstrukte mit bisherigen Erkenntnissen zum Entwicklungsaufwand sowie zu den Erfolgsfaktoren und Risiken für Integrationsvorhaben macht ferner deutlich, dass mit der Untersuchung viele neuartige Einblicke in das Verständnis erfahrener Praktiker gewonnen werden konnten. Durch die Dokumentierung der beiden gegenüberstehenden Pole der Konstrukte im Wortlaut der Untersuchungsteilnehmer konnten dabei reichhaltige Informationen gewonnen werden. Sie repräsentieren die Interpretationsmuster der Untersuchungsteilnehmer besser als wenn lediglich punktuelle Bezeichnungen für die vermuteten Einflussfaktoren erfragt worden wären.⁶¹⁶

Schließlich war es das wissenschaftliche Erkenntnisziel der Untersuchung, geeignete Grundlagen für die Generierung empirischer Hypothesen zu schaffen. Empirische Hypothesen wurden definiert als Aussagen bezüglich einer angenommenen Beziehung zwischen einem oder mehreren unabhängigen Attributen von Entitäten und einem ab-

⁶¹⁶ Im Sinne der Theorie der persönlichen Konstrukte bilden die Konstruktpole die Extrema von Skalen, anhand derer die Merkmalsausprägungen von Integrationsvorhaben beurteilt werden können. Sie weisen damit bereits ein grundlegendes Merkmal verbreiteter Verfahren zur Aufwandschätzung von Softwareentwicklungsvorhaben auf, die ebenfalls die Beurteilung der Ausprägung von Einflussfaktoren anhand von Skalen erfordern.

hängigen Attribut. Sie repräsentieren das Verständnis eines Phänomens, dessen Gültigkeit durch eine empirische Studie überprüft werden soll.⁶¹⁷

Die erhobenen Konstrukte drücken jeweils eine angenommene Beziehung zwischen einem oder mehreren unabhängigen Merkmalen von Integrationsvorhaben und dem erforderlichen Entwicklungsaufwand aus. Sie repräsentieren das Verständnis der erfahrenen Untersuchungsteilnehmer und könnten demnach bereits als empirische Hypothesen im Sinne der genannten Definition angesehen werden.

Zusammenfassend kann die Frage, ob die vorliegende Untersuchung zu neuen und relevanten wissenschaftlichen Einsichten geführt hat, somit positiv beantwortet werden.

6.2.2 Perspektiven für die wissenschaftliche Verwendung der Untersuchungsergebnisse

Auch wenn die erhobenen Konstrukte bereits als empirische Hypothesen im Sinne der genannten Definition angesehen werden könnten, stellen die vorliegenden Untersuchungsergebnisse lediglich einen ersten, wenngleich notwendigen Schritt zu einem Verständnis der Einflussfaktoren auf den Entwicklungsaufwand von Integrationsvorhaben dar, dessen Überprüfung verlässliche Schlüsse auf die Realität erlauben würde. In weiterführenden Untersuchungen sollten die gewonnenen Erkenntnisse aufgegriffen und weiterentwickelt werden. Diesbezüglich werden im Weiteren einige Perspektiven aufgezeigt.

6.2.2.1 Priorisierung der Konstrukte und Untersuchung der Wirkungszusammenhänge

Es können drei grundlegende Empfehlungen für die Weiterentwicklung der Untersuchungsergebnisse gegeben werden: die Priorisierung der Konstrukte, die Bestimmung der anzunehmenden Wirkungszusammenhänge zwischen den beschriebenen Merkmalen von Integrationsvorhaben und die Bestimmung der anzunehmenden Wirkungsbeziehungen zu den Teilaufgaben der Anwendungsintegration.

⁶¹⁷ Vgl. Briand, Morasca, Basili /Measures/ 1114.

▪ **Priorisierung der Konstrukte**

Die erhobenen Konstrukte repräsentieren die persönlichen Erfahrungen der Untersuchungsteilnehmer. Dabei war es das Anliegen des idiographischen Forschungsansatzes, die wesentlichen Leitgedanken erfahrener Praktiker zu erfassen. Dagegen wurde jedoch nicht angestrebt, Aussagen über die Repräsentativität dieser Erfahrungswerte treffen zu können. Somit ist unklar, inwiefern die dokumentierten Erfahrungswerte von anderen Praktikern geteilt werden und welche Bedeutung sie ihnen beimessen.

In einer weiterführenden Untersuchung könnten die Konstrukte durch weitere erfahrene Praktiker beurteilt werden, um auf diese Weise diejenigen der 310 Konstrukte auszuwählen, bei denen von einer hohen Übertragbarkeit und großen Bedeutung für die Erklärung des Entwicklungsaufwands von Integrationsvorhaben ausgegangen werden kann. Diese Priorisierung und Auswahl von Konstrukten erscheint auch aus Gründen der Praktikabilität anschließender Untersuchungen empfehlenswert. Diese Vorgehensweise wird beispielsweise von El Emam und Madhavji sowie von Dyba für die Untersuchung der Erfolgsfaktoren der Anforderungsanalyse (requirements engineering) bzw. der Erfolgsfaktoren für die Verbesserung des Softwareentwicklungsprozesses (software process improvement) angewandt.⁶¹⁸

▪ **Wirkungszusammenhänge zwischen den Merkmalen**

Mit der vorliegenden Untersuchung konnten die persönlichen Erfahrungen der Untersuchungsteilnehmer explizit gemacht werden, worin wesentliche Einflussfaktoren auf den Entwicklungsaufwand bestehen. Es bleibt jedoch unklar, welche Beziehungen zwischen den einzelnen Einflussfaktoren anzunehmen sind. In gleicher Weise wie die Einflussfaktoren als Knotenpunkte in einem komplexen Netzwerk aus Ursachen und Wirkungen verstanden werden können, ist auch von Über- und Unterordnungsverhältnissen zwischen den erhobenen Konstrukten auszugehen.

In einer weiterführenden Untersuchung könnte daher durch erfahrene Praktiker beurteilt werden, welche Wirkungszusammenhänge zwischen den beschriebenen Merkmalen von Integrationsvorhaben anzunehmen sind. Durch eine Analyse dieser Wirkungszusammenhänge könnte die Identifizierung besonders wichtiger Knotenpunkte und damit eine

⁶¹⁸ Vgl. El Emam, Madhavji /Measuring the Success/ 205-208; Dyba /Software Process Improvement/ 368f.

weitere Priorisierung von Konstrukten erfolgen. Damit wäre es auch möglich, einzelne Merkmale auszuwählen, deren Wirkungszusammenhänge mit dem Entwicklungsaufwand empirisch überprüft werden sollen. Gleichzeitig würde ein besseres Verständnis davon gewonnen, welche Rahmenbedingungen (im Sinne der übrigen Merkmale) bei dieser Überprüfung kontrolliert werden müssen.

Als geeignete Vorgehensweise erscheint diesbezüglich die Methode des Causal Mapping, mit der Aussagen von Auskunftspersonen zu Wirkungszusammenhängen erhoben und analysiert werden können. Sie wird beispielsweise von Nelson u. a. für die Untersuchung des Konzepts der Expertise in der Betreuung von Anwendungssystemen (software operation support expertise) angewandt.⁶¹⁹

▪ **Wirkungsbeziehungen zu den Teilaufgaben der Anwendungsintegration**

Wie erläutert resultiert die Arbeitslast eines Integrationsvorhabens aus den zu erfüllenden Entwicklungs- und Unterstützungsaufgaben. Somit kann der Entwicklungsaufwand durch die Betrachtung einzelner Teilaufgaben oder der mit ihnen verbundenen Arbeitsergebnisse strukturiert werden.

Während die erhobenen Konstrukte insbesondere ein besseres Verständnis der unabhängigen Attribute der angestrebten empirischen Hypothesen ermöglichen, leisten sie keinen differenzierten Erkenntnisbeitrag zum abhängigen Attribut, dem Entwicklungsaufwand. Eine Aufwandschätzung zur Schaffung einer zuverlässigen Planungsgrundlage für Integrationsvorhaben erfordert jedoch auch die Kenntnis der Bedeutung der identifizierten Einflussfaktoren für die einzelnen Teilaufgaben.

Da gegenwärtig lediglich einzelne Ansätze zur Strukturierung von Integrationsvorhaben existieren und im Unterschied zur klassischen Softwareentwicklung keine umfassenden und allgemein akzeptierten Vorgehensmodelle der Anwendungsintegration zur Verfügung stehen, sollten auch diesbezüglich weitere Untersuchungen durchgeführt werden. Beispielsweise könnte durch die Beobachtung von Integrationsvorhaben und begleitende Interviews der Projektbeteiligten ein besseres Verständnis von den mit den Teilaufgaben der Anwendungsintegration verbundenen Tätigkeiten und Arbeitsergebnissen gewonnen werden. Diese Vorgehensweise wurde beispielsweise im Zusammenhang mit der Konzeption des COCOMO II-Verfahrens und der darauf aufbauenden COCOTS-

⁶¹⁹ Vgl. Nelson u. a. /Software Operations Support Expertise/.

und COSOSIMO-Verfahren angeführt.⁶²⁰ Darauf aufbauend könnte wiederum über die Methode des Causal Mapping untersucht werden, welche Wirkungsbeziehungen zwischen den Einflussfaktoren und den differenzierten Komponenten des Entwicklungsaufwands anzunehmen sind. Auf diese Weise könnten präzisere empirische Hypothesen im Hinblick auf das abhängige Attribut formuliert werden.

6.2.2.2 Präzisierung der Konstrukte

Gemäß Briand, Morasca und Basili schließt sich der Generierung empirischer Hypothesen die Definition von Maßen für die betrachteten Attribute, die Konkretisierung der Hypothesen aufgrund der Maße und ihre empirische Überprüfung an.⁶²¹ Im Zuge dieser Schritte ist eine Präzisierung der erhobenen Konstrukte erforderlich. Diese Notwendigkeit wird in seiner Bedeutung oftmals übersehen oder unterschätzt. In der Folge werden auf die erhobenen Informationen oftmals unzulässige Analysemethoden angewandt und falsche Schlüsse auf die Realität gezogen.⁶²² Die erforderliche Präzisierung der Konstrukte wird daher im Vergleich zu den vorangehenden Empfehlungen für weiterführende Untersuchungen besonders detailliert erläutert.

Gemäß Rossiter muss ein Konstrukt im Hinblick auf das betroffene Objekt, das betrachtete Attribut und die Beurteiler definiert sein, um geeignete Hypothesen für die Entwicklung von Maßen aufstellen zu können.⁶²³ Die in der vorliegenden Untersuchung erhobenen Konstrukte scheinen diese Forderung zu erfüllen.

Bislang wurde vereinfachend davon gesprochen, dass die in der vorliegenden Untersuchung erhobenen Konstrukte Merkmale von Integrationsvorhaben beschreiben. Präziser ausgedrückt beschreiben sie jeweils zwei entgegengesetzte Ausprägungen von Attributen bestimmter Objekte in Integrationsvorhaben. Beispielsweise beschreibt das Konstrukt „Die Komplexität eines zu unterstützenden Geschäftsprozesses ist gering – Die Komplexität eines zu unterstützenden Geschäftsprozesses ist groß“ das Merkmal

⁶²⁰ Vgl. Abts, Boehm, Clark /COCOTS/ 2f.; Abts, Boehm /COCOTS/ 19-22.

⁶²¹ Vgl. diesbezüglich Briand, Morasca, Basili /Measures/ bzw. Kapitel 1.1.3.

⁶²² Vgl. Rossiter /Scale Development/ 305-308.

⁶²³ Die weiteren Ausführungen in diesem Unterkapitel beziehen sich auf die von Rossiter vorgeschlagene C-OAR-SE-Methode zur Entwicklung von Maßen (vgl. Rossiter /Scale Development/).

„Komplexität“ des Objekts „zu unterstützender Geschäftsprozess“ anhand der entgegengesetzten Ausprägungen „gering“ und „groß“.

Die Konstrukte spiegeln ferner die Sichtweise der Untersuchungsteilnehmer wider. Die Einschätzung der Merkmalsausprägungen basiert demnach auf der Betrachtungsperspektive der für die Entwicklung und Bereitstellung der Integrationslösung zuständigen Personen. Im Falle des angeführten Beispiels bedeutet dies, dass die Komplexität eines zu unterstützenden Geschäftsprozesses durch die Entwickler der Integrationslösung und nicht durch die Benutzer der zu integrierenden Anwendungssysteme beurteilt werden sollte.

Die erhobenen Konstrukte schaffen damit eine wichtige Grundlage für die Generierung empirischer Hypothesen. Gemäß Rossiter erfordert die Entwicklung von Maßen für die betrachteten Attribute jedoch eine weitergehende konzeptionelle Definition eines Konstrukts: die Klassifizierung des Objekts, die Klassifizierung des Attributs und die Konkretisierung der Beurteiler.

▪ **Klassifizierung des Objekts**

Das Objekt eines Konstrukts kann entweder singulär sein, ein Kollektiv von Einzelobjekten darstellen oder aus verschiedenen Komponenten bestehen. Des Weiteren kann das Objekt konkret oder abstrakt sein. Anhand dieser beiden Dimensionen differenziert Rossiter drei alternative Klassen von Objekten: konkrete und singuläre Objekte (concrete singular objects), abstrakte und kollektive Objekte (abstract collective objects) sowie abstrakte und formative Objekte (abstract formed objects). Jede dieser Klassen erfordert ein spezifisches Vorgehen bei der Erhebung und Analyse von zugehörigen empirischen Daten.⁶²⁴

Im Falle konkreter und singulärer Objekte ist davon auszugehen, dass alle Beurteiler bewusst ein und dasselbe Objekt betrachten. Im genannten Beispielsfall würde dies bedeuten, dass ein zu unterstützender Geschäftsprozess ganzheitlich von den Entwicklern der Integrationslösung hinsichtlich seiner Komplexität beurteilt wird.

Abstrakte und kollektive Objekte werden von den Beurteilern als unterschiedlich wahrgenommen, d. h. sie werden als eine Menge eigenständiger Einzelobjekte angesehen,

⁶²⁴ Zur Klassifizierung des Objekts vgl. im Weiteren Rossiter /Scale Development/ 309-313.

die jedoch auf übergeordneter Ebene eine Einheit bilden. Wäre das angeführte Beispielkonstrukt wie folgt formuliert: „Die Komplexität der zu unterstützenden Geschäftsprozesse ist gering – Die Komplexität der zu unterstützenden Geschäftsprozesse ist groß“, so könnten die Entwickler der Integrationslösung beispielsweise in Abhängigkeit von den jeweils angestrebten Leistungen verschiedene Typen von Geschäftsprozessen differenzieren (in diesem Sinne unterschied einer der Untersuchungsteilnehmer zwischen kaufmännischen Prozessen und Servicemanagementprozessen).

Abstrakte und formative Objekte werden aus Sicht der Beurteiler aus mehreren Komponenten gebildet. Wäre das angeführte Beispielkonstrukt wie folgt formuliert: „Die Komplexität der zu entwickelnden Integrationslösung ist gering – Die Komplexität der zu entwickelnden Integrationslösung ist groß“, so könnten die Entwickler unter der Integrationslösung zum einen eine Kombination von Softwaresystemen und zugehörigen Daten verstehen. Zum anderen könnte das Konzept der Integrationslösung für sie beinhalten, dass ein bestimmtes Integrationsproblem tatsächlich gelöst wird. Im Unterschied zu abstrakten und kollektiven Objekten, die aus Einzelobjekten gebildet werden, vereinen abstrakte und formative Objekte somit verschiedene Dimensionen in sich. Die Komplexität der Integrationslösung würde aus Sicht der Entwickler aus der Komplexität ihres Aufbaus (d. h. der Integrationsarchitektur) und der Komplexität der ihr zugrunde liegenden Problemstellung folgen.

Die Klassifizierung des Objekts eines Konstruktes ist von zentraler Bedeutung für dessen Messung: ein konkretes und singuläres Objekt kann in einem einzelnen Item erfasst werden während die Messung eines abstrakten und kollektiven Objekts oder eines abstrakten und formativen Objekts auf einer spezifischen Kombination der Messungen der Einzelobjekte bzw. der einzelnen Dimensionen basiert. Diese Zusammenhänge müssen auch bei der Analyse der erhobenen Daten berücksichtigt werden.⁶²⁵

⁶²⁵ Zur Erhebung und Analyse von Informationen zu Objekten der verschiedenen Klassen vgl. ebenfalls Rossiter /Scale Development/ 309-313.

▪ **Klassifizierung des Attributs**

Das Attribut eines Konstrukts kann entweder konkret und singulär (concrete singular), abstrakt und formativ (abstract formed) oder abstrakt und reflexiv (abstract eliciting) sein.⁶²⁶

Im Falle konkreter und singulärer Attribute ist wie bei der entsprechenden Klasse von Objekten davon auszugehen, dass alle Beurteiler bewusst ein und dasselbe Attribut ganzheitlich einschätzen. Als Beispiel kann das erhobene Konstrukt „Die Anzahl der zu unterstützenden Geschäftsprozesse ist gering – Die Anzahl der zu unterstützenden Geschäftsprozesse ist groß“ angeführt werden, bei der die Anzahl durch die Entwickler der Integrationslösung ganzheitlich beurteilt wird.

Abstrakte und formative Attribute werden wie die entsprechende Klasse von Objekten durch verschiedene Dimensionen gebildet. Diesbezüglich kann die Komplexität der zu unterstützenden Geschäftsprozesse als Beispiel angeführt werden, die durch die Entwickler beispielsweise als Resultat der Anzahl der Ereignisse und Aktivitäten, der Schwierigkeit der Aktivitäten und der Menge der Beziehungen zu anderen Geschäftsprozessen beurteilt werden könnte.

Abstrakte und reflektive Attribute haben einen entgegengerichteten Charakter: nicht verschiedene Dimensionen bilden das Attribut, sondern die Ausprägung des Attributs kommt in verschiedener Weise zum Ausdruck. Das erhobene Konstrukt „Unter den Projektbeteiligten herrscht Teamgeist – Unter den Projektbeteiligten herrscht kein Teamgeist“ kann diesbezüglich als Beispiel angeführt werden. Wenn die Entwickler der Integrationslösung den Teamgeist als gegeben sehen, so könnte dies aus ihrer Sicht in verschiedenen Verhaltensweisen der Projektbeteiligten zum Ausdruck kommen, wie beispielsweise einem freundlichen Umgang miteinander, großer Hilfsbereitschaft oder einer gemeinsamen Identifizierung mit Arbeitsergebnissen.

Die Klassifizierung des Attributs eines Konstruktes ist wiederum von zentraler Bedeutung für dessen Messung: Kann es in einem Item oder muss es in mehreren Items erfasst werden? Wie müssen die erhobenen Informationen ausgewertet werden? Beispielsweise verbietet sich der Ausschluss einzelner Items zu einem abstrakten und formativen Attribut aufgrund geringer Korrelationen zwischen den Beurteilungen. Dagegen empfiehlt

⁶²⁶ Zur Klassifizierung des Attributs vgl. im Weiteren Rossiter /Scale Development/ 313-318.

sich dieses Vorgehen im Falle abstrakter und reflektiver Attribute, um besonders aussagekräftige Indikatoren zu bestimmen.⁶²⁷

Ein Beispiel für die Verletzung dieser Regeln wurde in dieser Arbeit bereits angesprochen: der erläuterte Beitrag zu den Einflussfaktoren auf den Erfolg der Anwendungsintegration.⁶²⁸ Darin verweisen die Autoren darauf, dass ein Zusammenhang zwischen der Zufriedenheit der Anwender und dem Erfolg der Anwendungsintegration statistisch nicht nachgewiesen werden konnte. Die gewählte Analysemethode widerspricht jedoch der früheren Aussage, dass der Erfolg der Anwendungsintegration aus dem Erreichungsgrad der unterschiedenen Komponenten resultiert, dass es sich also um ein abstraktes und formatives Attribut handelt.⁶²⁹ Auch die Berechnungen, welcher Anteil der Varianzen der Komponenten durch die abstrakten und formativen Attribute erklärt werden, erscheinen nicht als sinnvoll. Diese Analysemethode hätte vielmehr im Falle abstrakter und reflektiver Attribute gewählt werden sollen.⁶³⁰ Letztlich erscheinen die von den Autoren gezogenen Schlüsse zu den Erfolgsfaktoren der Anwendungsintegration damit als inkonsistent und ungültig.

▪ **Konkretisierung der Beurteiler**

Schließlich müssen auch die Beurteiler eines Konstrukts genau definiert werden, denn die Einschätzung von Objekten und Attributen hängt i. d. R. von der Betrachtungsperspektive ab.⁶³¹

In diesem Zusammenhang ist zum einen festzulegen, ob es sich bei den Beurteilern um einzelne Individuen handelt, die unabhängig voneinander Objekte und Attribute einschätzen, oder ob die Beurteilung durch eine Gruppe von Personen erfolgen soll. Beispielsweise könnte die Komplexität eines zu unterstützenden Geschäftsprozesses durch einen einzelnen Entwickler der Integrationslösung beurteilt werden. Um Rückschlüsse auf den gesamten Entwicklungsaufwand aller Projektmitarbeiter ziehen zu können,

⁶²⁷ Zur Erhebung und Analyse von Informationen zu Attributen der verschiedenen Klassen vgl. ebenfalls Rossiter /Scale Development/ 313-318.

⁶²⁸ Vgl. Klesse, Wortmann, Schelp /Applikationsintegration/ bzw. Kapitel 2.3.3.1.

⁶²⁹ „Der Erfolg der Applikationsintegration bemisst sich also daran, ob das Unternehmen die selbstgesetzten Ziele bezüglich der Applikationsintegration erreicht hat.“ (Klesse, Wortmann, Schelp /Applikationsintegration/ 260)

⁶³⁰ Vgl. Rossiter /Scale Development/ 314f.

⁶³¹ Zur Konkretisierung der Beurteiler vgl. im Weiteren Rossiter /Scale Development/ 318f.

könnte es jedoch auch angebracht sein, zu erfassen, wie die Komplexität des Geschäftsprozesses insgesamt in der Gruppe der Entwickler eingeschätzt wird.

Neben der Differenzierung ob Einzelurteile oder Gruppenurteile erfasst werden sollen, können weitere Merkmale der Beurteiler festgelegt werden, z. B. im Hinblick auf den Erfahrungshintergrund, wie dies bei der Auswahl der Auskunftspersonen für die vorliegende Untersuchung geschehen ist.

▪ **Präzisierung der Konstrukte als eigenständige Untersuchung**

Bei der beispielhaften Erläuterung der vorangehenden Ausführungen wurden die Objekte und Attribute einiger erhobener Konstrukte durch den Autor dieser Arbeit klassifiziert. Im Prozess der wissenschaftlichen Erkenntnis sollte es jedoch prinzipiell vermieden werden, die Untersuchungsergebnisse auf diese Weise zu interpretieren. Gemäß Rossiter erfordert die Klassifizierung vielmehr die Einbeziehung einer repräsentativen Teilmenge der Beurteiler. Er empfiehlt deren Verständnis von den Objekten und Attributen in offenen Interviews systematisch zu erheben und daraus die geeignete Klassifizierung abzuleiten.⁶³² Die Präzisierung der erhobenen Konstrukte kann daher als eigenständige Untersuchung angesehen werden, die über den Rahmen dieser Arbeit hinausgeht.

Erste Anregungen zur Klassifizierung der Attribute und zur Definition von Maßen können jedoch bereits gegeben werden. Beispielhaft konnten anhand der Transkripte verschiedene Erläuterungen der Untersuchungsteilnehmer identifiziert werden, die auf abstrakte und formative oder abstrakte reflektive Attribute hindeuten. Sie werden in Tabelle 6-1 aufgeführt.

⁶³² Vgl. Rossiter /Scale Development/ 313-318.

Attribut	Erläuterung
Hinweise auf abstrakte und formative Attribute	
Qualität einer verwendeten Schnittstelle	Ein Teilnehmer erläuterte, dass eine geringe Qualität einer Schnittstelle insbesondere in einer fehlenden Trennung von Schichten, wie Repräsentation und Protokoll, oder in der Vermischung von fachlichen und technischen Aspekten zum Ausdruck komme.
Integrationsumfang	Ein Teilnehmer erläuterte, dass der Integrationsumfang beispielsweise mit der Anzahl der zu integrierenden Anwendungssysteme, der Notwendigkeit im Rahmen der Integrationslösung eine neue Infrastruktur aufzubauen oder der Anzahl der Schnittstellen zunehme.
Komplexität der Integrationsarchitektur	Ein Teilnehmer gab an, dass die Komplexität einer Integrationsarchitektur mit zunehmender Vielfalt der eingesetzten Integrationsprodukte und Systemplattformen wachse.
Komplexität eines zu entwickelnden Moduls	Ein Teilnehmer erläuterte, dass die Komplexität eines Moduls von der verwendeten Technologie und dem verfolgten Entwicklungsparadigma abhänge.
Komplexität eines zu integrierenden Anwendungssystems	Ein Teilnehmer erläuterte, dass die Komplexität eines Anwendungssystems beispielsweise aus der Komplexität der Funktionen, Schnittstellen und Daten resultiere.
Komplexität eines zu unterstützenden Geschäftsprozesses	Ein Teilnehmer erläuterte, dass die Komplexität der Prozessschritte beispielsweise mit der Anzahl der damit verbundenen Use Cases oder Funktionen zunehme.
Komplexität eines zu unterstützenden Use Case	Ein Teilnehmer erläuterte, dass eine hohe Komplexität eines Use Case beispielsweise daraus resultiere, dass er die Verwendung mehrerer Anwendungssysteme umfasst, er "über Systemgrenzen hinweg geht" oder mehrere Funktionen erfordert.
Komplexität einer zu realisierenden Funktion	Ein Teilnehmer erklärte, dass die Komplexität einer Funktion beispielsweise mit der Menge der Parameter und der Kompliziertheit der erforderlichen Berechnungen zunehme.
Komplexität einer Schnittstelle	Die Komplexität einer Schnittstelle wurde durch die Teilnehmer unterschiedlich erläutert. Einer unterschied Schnittstellen auf Datenebene und Schnittstellen auf Funktionsebene. Die Komplexität einer Schnittstelle auf Datenebene nehme beispielsweise mit der Anzahl der Funktionsaufrufe zu. Ein weiterer Teilnehmer gab an, dass die Komplexität einer Schnittstelle beispielsweise mit der Anzahl zu unterstützender Versionen bzw. erforderlicher Backward-Kompatibilität, der qualitätsbezogenen Anforderungen, wie z. B. der Verfügbarkeit, oder der Anzahl der Parameter zunehme. Schließlich erläuterte ein weiterer Teilnehmer, dass beispielsweise eine Schnittstelle "auf Bits-und-Byte-Basis" wesentlich komplexer als ein textbasierter Nachrichtenaustausch sei.
Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht	Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht wurden durch die Teilnehmer ebenfalls unterschiedlich erläutert. Dabei lassen die Erläuterungen sowohl auf ein abstraktes und zusammengesetztes Attribut als auch auf ein abstraktes und begründendes Attribut schließen. Im ersten Sinne gab einer der Teilnehmer an, dass die Komplexität der Aufgabenstellung beispielsweise mit der Komplexität der zu unterstützenden Geschäftsprozesse oder den zu behandelnden Ausnahmen zunehme. In gleicher Weise erläuterte ein anderer Teilnehmer, dass die Komplexität der Aufgabenstellung beispielsweise mit der Anzahl der fachlichen Funktionalitäten, der Vielfalt der zu berücksichtigenden Sonderfälle und Ausnahmen zunehme.

Attribut	Erläuterung
Komplexität eines Adapters	Ein Teilnehmer erläuterte anhand eines Beispiels, dass ein komplexer Adapter aus verschiedenen Komponenten bestehen könne, verschiedene eigene Parameter erfordere oder eine eigene Benutzerschnittstelle besitzen könne. Letztlich handele es sich dabei oftmals um eigenständige Softwaresysteme, die aber als Adapter bezeichnet und als solche eingesetzt würden.
Komplexität einer zu realisierenden Transformation ("Mapping")	Ein Teilnehmer erläuterte, dass eine komplexe Transformation beispielsweise viele Verarbeitungsschritte und den Zugriff auf weitere Softwaresysteme erforderlich mache.
Sicherheitsanforderungen an die Integrationslösung	Ein Teilnehmer erläuterte, dass sich die Sicherheitsanforderungen beispielsweise auf Transport, Datenintegrität oder Verschlüsselung beziehen könnten.
qualitätsbezogene/nicht-funktionale Anforderungen an die Integrationslösung	Ein Teilnehmer erläuterte, dass sich die nicht-funktionalen Anforderungen beispielsweise auf die Performanz oder die Service Level 24x7 beziehen.
Komplexität einer Datenbanktabelle	Ein Teilnehmer gab an, dass die Komplexität einer Datenbanktabelle beispielsweise mit Bitemporalität (Datenänderungen müssen nachgehalten werden) oder der Anzahl der Datenelemente zunehme.
Komplexität des Datenmodells	Ein Teilnehmer erläuterte, dass die Komplexität des Datenmodells beispielsweise auf der Anzahl der Dimensionen des Datenwürfels im Sinne eines Data-Warehouse basiere.
Komplexität eines Datenelements	Ein Teilnehmer erläuterte, dass die Komplexität eines Datenelements mit der Anzahl der zu verwaltenden Parameter zunehme.
Hinweise auf abstrakte und reflektive Attribute	
quantitative Interpretation der Anforderungen an die Qualität der Integrationslösung	Ein Teilnehmer nannte als Ausdruck für eine quantitative Interpretation der Anforderungen an die Qualität der Integrationslösung die Vorgabe maximal zulässiger Häufigkeiten bestimmter abstrakter Fehlerarten bei der Abnahme.
Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht	Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht wurden durch die Teilnehmer unterschiedlich erläutert. Dabei lassen die Erläuterungen sowohl auf ein abstraktes und zusammengesetztes Attribut als auch auf ein abstraktes und begründendes Attribut schließen. Im letzteren Sinne gab einer der Teilnehmer an, dass ein größerer fachlicher Umfang der Aufgabenstellung im Allgemeinen in einem umfangreicheren Dokument des Fachkonzepts oder Lastenhefts zum Ausdruck komme.
Komplexität von Informationsstrukturen	Ein Teilnehmer erläuterte, dass komplexe Informationsstrukturen beispielsweise darin zum Ausdruck kämen, dass komplizierte Baumstrukturen hinsichtlich der Informationen bestehen.

Tabelle 6-1: Hinweise auf die Klassifizierung der Attribute in einigen der erhobenen Konstrukte

6.3 Praktische Relevanz der Untersuchungsergebnisse

Die gewonnenen Untersuchungsergebnisse sind nicht nur für die wissenschaftliche Auseinandersetzung mit den Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration von Bedeutung. Vielmehr liefern sie bereits wertvolle Anhaltspunkte für die praktische Aufwandschätzung von Integrationsvorhaben: die explizit gemachten persönlichen Erfahrungen der Untersuchungsteilnehmer können als Checkliste bei der Schätzung des erforderlichen Entwicklungsaufwands eingesetzt werden.

Die Aufwandschätzung von Entwicklungsvorhaben kann in zwei Komponenten unterschieden werden, eine vorgehensbezogene Komponente und eine inhaltliche Komponente, die jeweils durch spezifische Checklisten unterstützt werden können.

▪ Vorgehensbezogene Checklisten für die Aufwandschätzung

Die vorgehensbezogene Komponente der Aufwandschätzung bezieht sich darauf, welche Aktivitäten von welchen Personen durchgeführt werden, um zu einem Schätzergebnis zu gelangen.

Dabei gilt es zum einen, die negativen Auswirkungen der begrenzten Rationalität, der unvollkommenen menschlichen Interpretation der Realität sowie der unvollständigen oder verzerrten Erinnerung an vergangene Ereignisse auf die Zuverlässigkeit der Schätzergebnisse entgegenzuwirken. Zum anderen gilt es, dem politischen Aspekt der Aufwandschätzung Rechnung zu tragen. Untersuchungen, wie beispielsweise eine von Lederer u. a. durchgeführte Fallstudie, weisen darauf hin, dass die konfliktären Interessen der Beteiligten zu einer Instrumentalisierung der Schätzung des Entwicklungsaufwands führen können, d. h. dass nicht zuverlässige, sondern den persönlichen Interessen entsprechende Schätzergebnisse angestrebt werden (z. B. könnte der Entwicklungsaufwand bewusst niedriger angesetzt werden als erwartet, um auf diese Weise eine höhere Wirtschaftlichkeit des Entwicklungsvorhabens auszuweisen und die Auftragsvergabe durch den Auftraggeber sicherzustellen).⁶³³ Einer solchen Instrumentalisierung zulasten der Zuverlässigkeit muss ebenfalls durch ein geeignetes Vorgehen im Zuge der Aufwandschätzung entgegengewirkt werden.

⁶³³ Zur politischen Instrumentalisierung der Aufwandschätzung vgl. Lederer u. a. /Cost Estimating/ 161f.; Robey, Markus /Rituals/ 9f.;

Es existieren verschiedene Vorschläge für Checklisten zur Unterstützung der vorgehensbezogenen Komponente der Aufwandschätzung, beispielsweise von Park sowie von Jorgensen und Molokken.⁶³⁴ Die von ihnen angeführten Empfehlungen erscheinen von grundlegender Bedeutung für Entwicklungsvorhaben und sind damit auch für Integrationsvorhaben relevant.

Park empfiehlt die Ergebnisse der Aufwandschätzung anhand von sieben Fragen zu beurteilen, zu deren Beantwortung er jeweils verschiedene Kriterien vorgibt:⁶³⁵

- Sind die Ziele der Aufwandschätzung präzise und korrekt definiert?
- Wurden die geschätzten Aufgaben angemessen hinsichtlich ihres Umfangs beurteilt?
- Sind die Schätzergebnisse konsistent mit dem benötigten Entwicklungsaufwand früherer Projekte?
- Wurden alle relevanten Einflussfaktoren identifiziert und erörtert?
- Wurden geeignete Maßnahmen ergriffen, um die Integrität der Aufwandschätzung sicherzustellen?
- Sind die verfügbaren Informationen zu vorangehenden Projekten geeignet, um zuverlässige Aufwandschätzungen zu unterstützen?
- Haben sich die herrschenden Bedingungen seit der Durchführung der Aufwandschätzung verändert?

Jorgensen und Molokken unterscheiden dagegen vier Phasen der Aufwandschätzung die insgesamt 12 Aktivitäten umfassen. Zu diesen Aktivitäten führen sie im Weiteren verschiedene Empfehlungen auf, an denen sich das Vorgehen orientieren sollte (vgl. Tabelle 6-2).⁶³⁶

⁶³⁴ Vgl. Park /Checklist/; Jorgensen, Molokken /Checklist/.

⁶³⁵ Vgl. im Folgenden Park /Checklist/ 1.

⁶³⁶ Vgl. im Folgenden Jorgensen, Molokken /Checklist/ 134.

Phase der Aufwandschätzung	Aktivitäten
Vorbereitungsphase	1. Verstehen des Schätzproblems 2. Einigung auf Entscheidungen und Annahmen auf denen die Schätzung aufbaut 3. Sammlung relevanter Informationen für die Aufwandschätzung 4. Gestaltung des Schätzprozesses
Schätzphase	5. Schätzung des mit größter Wahrscheinlichkeit erforderlichen Entwicklungsaufwands 6. Beurteilung der Unsicherheit des Schätzergebnisses 7. Review des Schätzprozesses
Anwendungsphase	8. Anwendung des Schätzergebnisses in der Angebotserstellung 9. Anwendung des Schätzergebnisses in der Planung und Budgetierung 10. Kommunizierung des Schätzergebnisses 11. Überwachung des Entwicklungsaufwands
Lernphase	12. Identifizierung von Verbesserungsmöglichkeiten für zukünftige Schätzungen

Tabelle 6-2: Phasen und Aktivitäten der Aufwandschätzung gemäß Jorgensen und Molokken⁶³⁷

▪ Inhaltliche Checklisten für die Aufwandschätzung

Neben der vorgehensbezogenen Komponente der Aufwandschätzung kann eine inhaltliche Komponente abgegrenzt werden, d. h. welche Einflussfaktoren in welcher Weise bei der Ermittlung des Schätzergebnisses berücksichtigt werden. Park sowie Jorgensen und Molokken weisen in diesem Zusammenhang lediglich darauf hin, dass die relevanten Einflussfaktoren identifiziert und erörtert bzw. relevante Informationen für die Aufwandschätzung gesammelt werden sollen. Die von ihnen angeführten vorgehensbezogenen Empfehlungen müssen daher um inhaltliche Anhaltspunkte ergänzt werden. Im Zuge dieser Arbeit wurde diesbezüglich verdeutlicht, dass dabei den Charakteristika unterschiedlicher Typen von Entwicklungsvorhaben Rechnung getragen werden muss.

Die vorliegenden Untersuchungsergebnisse liefern konkrete Hinweise, welche Einflussfaktoren aus der Sicht erfahrener Praktiker für den Entwicklungsaufwand der Anwendungsintegration von Bedeutung sind. Sie wurden in eine Checkliste überführt, die praktische Hilfe für die Erstellung und die Beurteilung von Schätzergebnissen bietet (siehe Anhang B).

⁶³⁷ Vgl. Jorgensen, Molokken /Checklist/ 134.

Gemäß Jorgensen ist der Nutzen solcher Checklisten für die Aufwandschätzung unumstritten und kann anhand verschiedener Beobachtungen begründet werden.⁶³⁸

Er verweist auf verschiedene Untersuchungen, die belegen, dass Checklisten auch erfahrene Aufwandschätzer unterstützen, indem sie wichtige Einflussfaktoren in Erinnerung rufen, die unter anderen Umständen übersehen worden wären. Insbesondere können Checklisten dazu anregen, auch solche Einflussfaktoren bei der Schätzung zu berücksichtigen, zu deren Ausprägung den Aufwandschätzern noch keine Informationen vorliegen. Des Weiteren führt Jorgensen empirische Erkenntnisse an, gemäß denen unerfahrene Aufwandschätzer durch den Einsatz von Checklisten gleichwertige Ergebnisse wie erfahrene Aufwandschätzer erzielen können. Er schließt mit der Feststellung, dass Checklisten die Dokumentierung der Informationen und Annahmen unterstützen, die einer Aufwandschätzung zugrunde gelegt werden. Die Schätzungen können somit intersubjektiv nachvollzogen und die Schätzergebnisse differenziert dem tatsächlich entstandenen Entwicklungsaufwand gegenüber gestellt werden. Auf diese Weise können Checklisten einen wertvollen Beitrag zu einer kontinuierlichen Verbesserung der Schätzgenauigkeit leisten.

Auf diese Weise leistet die vorliegende Arbeit einen unmittelbaren Beitrag zur Lösung des aufgezeigten Praxisproblems: sie bietet erstmalig eine spezifische Unterstützung zur Schätzung des Entwicklungsaufwands der Anwendungsintegration.

⁶³⁸ Die weiteren Ausführungen zum Nutzen von Checklisten für die Aufwandschätzung basieren auf Jorgensen /Expert Estimation/ 51 sowie den dort angeführten Quellen.

7 Fazit

Die vorliegende Arbeit widmet sich den Einflussfaktoren auf den Entwicklungsaufwand der Anwendungsintegration. Ausgehend von der erläuterten Problemstellung und dem postulierten Erkenntnisziel wird ein umfangreicher konzeptioneller Bezugsrahmen entwickelt, an dem der bisherige Mangel an geeigneten Erkenntnissen verdeutlicht werden kann. Mit der daraufhin durchgeführten Untersuchung wurden differenzierte Einblicke in die persönlichen Erfahrungen und Ansichten kompetenter Praktiker genommen. Somit werden interessierten Wissenschaftlern eine Basis für weiterführende Untersuchungen und Praktikern wertvolle Anhaltspunkte für eine zuverlässige Aufwandschätzung von Integrationsvorhaben geliefert.

Der Wert der gewonnenen Erkenntnisse reicht dabei über die zugrunde gelegte Problemstellung hinaus, denn viele der von den Untersuchungsteilnehmern genannten Einflussfaktoren erweisen sich als gestaltbar. Die vorliegende Arbeit liefert somit vielfältige Anregungen für ein effektives Management von Integrationsvorhaben, bei dem gezielt Maßnahmen ergriffen werden, um vermeidbaren Entwicklungsaufwand zu vermeiden und den Erfolg der Integrationsvorhaben sicherzustellen.

Literaturverzeichnis

Abdel-Hamid, Madnick /Dynamics/

Tarek K. Abdel-Hamid; Stuart E. Madnick: Lessons Learned from Modelling the Dynamics of Software Development. In: Communications of the ACM. Nr. 12, Jg. 32, 1989, S. 1426-1455.

Abts, Boehm /COCOTS/

Christopher M. Abts; Barry W. Boehm: COTS Software Integration Cost Modeling Study. USC Technical Report, June 1997. University of Southern California, Center for Software Engineering. Los Angeles, 1997.

Agarwal u.a. /Estimating/

R. Agarwal; Manish Kumar; Yogesh S. Mallick; R. M. Bharadwaj; D. Anantwar: Estimating software projects. In: Software Engineering Notes. Nr. 4, Jg. 26, 2001, S. 60-67.

Albrecht /Productivity/

A. J. Albrecht: Measuring Application Development Productivity. In: Proceedings of the Joint IBM / SHARE / GUIDE Application Development Symposium, 14.-17. Oktober 1979. Monterey, CA 1979, S.83-92.

Alexander /EAI/

Sascha Alexander: EAI - Hoffnungsträger mit noch unklarer Zukunft. In: Computerwoche. Nr. 30, Jg.27, 2000, S. 9-10.

Altman /Integration Tools/

Ross Altman: Integrating Your Integration Tools? In: Business Integration Journal, Nr. 8, 2003, S. 36.

Andersson, Johnson /Integration Styles/

Jonas Andersson; Pontus Johnson: Architectural Integration Styles for Large-Scale Enterprise Software Systems. In: Fifth IEEE International Enterprise Distributed Object Computing Conference, Sep. 04.-07. 2001. Seattle, Washington 2001, S.224-236.

Antoniol, Fiutem, Lokan /Object-Oriented Function Points/

G. Antonioli; R. Fiutem; C. Lokan: Object-Oriented Function Points: An Empirical Validation. In: Empirical Software Engineering. Nr. 3, Jg. 8, 2003, S. 225-254.

Balzert /Software-Technik/

Helmut Balzert: Lehrbuch der Software-Technik: Software-Entwicklung. 2. Auflage, Heidelberg, Berlin 2000.

Basili, Caldiera, Rombach /GQM/

Victor R. Basili; Gianluigi Caldiera; H. Dieter Rombach: Goal Question Metric Paradigm. In: John J. Marcianiak (Hrsg.): Encyclopedia of Software Engineering. Volume 1. New York u. a. 1994, S. 528-532.

Basili, Weiss /Valid Software Engineering Data/

Victor R. Basili; David M. Weiss: A Methodology for Collecting Valid Software Engineering Data. In: IEEE Transactions On Software Engineering. Nr. 6, Jg. 10, 1984, S. 728-738.

Bass, Clements, Kazman /Software Architecture/

Len Bass; Paul Clements; Rick Kazman: Software Architecture in Practice. Reading, Mass. u. a. 1998.

Belanger /Taping Research Interviews/

France Belanger: Summary – Advice on Taping Research Interviews.
<http://www.qual.auckland.ac.nz/>, Abruf am 29.03.2006.

Boehm /Software Engineering Economics/

Barry W. Boehm: Software Engineering Economics. Englewood Cliffs, N. J. 1981.

Boehm, Fairley /Estimation/

Barry W. Boehm; Richard E. Fairley: Software Estimation Perspectives. In: IEEE Software. Nr. 6, Jg. 17, 2000, S. 22-26.

Boehm u. a. /COCOMO II/

Barry W. Boehm; Chris Abts; Winsor Brown; Sunita Chulani; Bradford K. Clark; Ellis Horowitz; Ray Madachy; Donald Reifer; Bert Steece: Software Cost Estimation with Cocomo II. Upper Saddle River, New Jersey 2000.

Boles, Friebe, Luhmann /Integrationsszenarien/

Cornelia Boles; Jörg Friebe; Till Luhmann: Typische Integrationsszenarien und deren Unterstützung durch Web Services und andere Technologien. In: Wilhelm Hasselbring; Manfred Reichert (Hrsg.): EAI 2004 - Enterprise Application Integration, Tagungsband des GI-/GMDS-Workshops EAI'04 OFFIS, 12.-13. Februar 2004. Oldenburg 2004, S.57-67.

Booth, Colomb, Williams /Research/

Wayne C. Booth; Gregory G. Colomb; Joseph M. Williams: The Craft of Research. Chicago 1995.

Boston Corporate Finance /Enterprise Integration/

Boston Corporate Finance: Enterprise Integration Software, 3/2003 Industry Spotlight, 2003.

Briand, Morasca, Basili /Measures/

Lionel C. Briand; Sandro Morasca; Victor R. Basili: An Operational Process for Goal-Driven Definition of Measures. In: IEEE Transactions On Software Engineering. Nr. 12, Jg. 28, 2002, S. 1106-1125.

Britton /IT-Architecture/

Chris Britton: IT Architectures and Middleware: Strategies for Building Large, Integrated Systems. Boston u. a. 2001.

Brodie, Stonebraker /Legacy Systems/

Michael L. Brodie; Michael Stonebraker: Migrating Legacy Systems: Gateways, Interfaces & The Incremental Approach. San Francisco, California 1995.

Brooks /Mythical Man Month/

Frederick P. Brooks: The mythical man month. Reading, Mass. u. a. 1995.

Buhl, Christ, Pape /Marktstudie/

Lothar Buhl; Jörg Christ; Ulrich Pape: Marktstudie: Softwaresysteme für Enterprise Application Integration. In: Wilhelm Dangelmaier; Markus Böhner(Hrsg.): ALB/HNI-Verlagsschriftenreihe, Band: 7. Paderborn 2001.

Bundschuh, Fabry /Aufwandschätzung/

Manfred Bundschuh; Axel Fabry: Aufwandschätzung von IT-Projekten. Bonn 2000.

Buschmann u. a. /Patterns/

Frank Buschmann; Regine Meunier; Hans Rohnert; Peter Sommerlad; Michael Stal: Pattern-Oriented Software Architecture. A System of Patterns. New York u. a. 1996.

Buxmann /Standards/

Peter Buxmann: Standards und Standardisierung. In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 434-435.

Chari, Seshadri /Integration/

Kaushal Chari; Saravanan Seshadri: Demystifying Integration. In: Communications of the ACM. Nr. 7, Jg. 47, 2004, S. 59-63.

Chen u. a. /Software Cost Modeling/

Zhihao Chen; Tim Menzies; Danie Port; Barry Boehm: Finding the Right Data for Software Cost Modeling. In: IEEE Software. Nr. 6, Jg. 22, 2005, S. 38-46.

Clements u. a. /Documenting Architectures/

Paul Clements; Felix Bachmann; Len Bass; David Garlan; James Ivers; Reed Little; Robert Nord; Judith Stafford: Documenting Software Architectures: Views and Beyond. Boston u. a. 2003.

Comella-Dorda /Legacy System/

Santiago Comella-Dorda; Kurt Wallnau; Robert C. Seacord; John Robert: A Survey of Legacy System Modernization Approaches: Technical Note CMU/SEI-2000-TN-003, Software Engineering Institute (SEI), Carnegie Mellon University, 2000.

Conte, Dunsmore, Shen /Metrics and Models/

S. D. Conte; H. E. Dunsmore; V. Y. Shen: Software Engineering Metrics and Models. Menlo Park, California u. a. 1986.

Costello /Deadline/

Scott H. Costello: Software Engineering Under Deadline Pressure. In: ACM Sigsoft Software Engineering Notes. Nr. 5, Jg. 9, 1984, S. 15-19.

Craig, Jassim /Management for IT/

Sue Craig; Hadi Jassim: People and project management for IT. Berkshire 1995.

CSE-USC /COCOMO II/

Center for Software Engineering (CSE): COCOMO II Model Manual.

<http://sunset.usc.edu/research/COCOMOII/index.html>. University of Southern California (USC), Center for Software Engineering (CSE). Los Angeles, CA, 1998. Abruf am 22.01.2006.

CSE-USC /Initial Survey/

N.N.: COSOSIMO Initial Survey, University of Southern California (USC), Center for Software Engineering (CSE). Los Angeles, LA, 2004.

<http://sunset.usc.edu/cse/pub/research/COSOSIMO/index.html>. Abruf am 25.01.2005.

Cummins /Enterprise Integration/

Fred A. Cummins: Enterprise Integration: An Architecture for Enterprise Application and Systems Integration. New York u. a. 2002.

Davenport /Enterprise System/

Thomas H. Davenport: Putting the Enterprise into the Enterprise System. In: Harvard Business Review. Nr. 4, Jg. 76, 1998, S. 121-131.

Davenport, Short /Business Process Redesign/

T. Davenport; J. Short: The New Industrial Engineering: Information Technology and Business Process Redesign. In: Sloan Management Review. Nr. 4, Jg. 31, 1990, S. 11-27.

DeMarco /Controlling/

Tom DeMarco: Controlling Software Projects. Upper Saddle River, New Jersey 1982.

DIN /DIN 69901/

DIN (Hrsg.): Projektwirtschaft; Projektmanagement; Begriffe. DIN 69901. Berlin 1987.

Dittrich, Gatzju, Geppert /ADBMS/

Klaus Dittrich; Stella Gatzju; Andreas Geppert: The Active Database Management System Manifesto: A Rulebase of ADBMS Features. In: ACM SIGMOD Record. Nr. 3, Jg. 25, 1996, S. 40-49.

Dresbach /Modelle/

Stefan Dresbach: Epistemologische Überlegungen zu Modellen in der Wirtschaftsinformatik. In: Jörg Becker; Wolfgang König; Reinhard Schütte; Oliver Wendt; Stephan Zelewski (Hrsg.): Wirtschaftsinformatik und Wirtschaftstheorie: Bestandsaufnahme und Perspektiven. Wiesbaden 1999, S. 71-94.

Dyba /Software Process Improvement/

Tore Dyba: An Instrument for Measuring the Key Factors of Success in Software Process Improvement. In: Empirical Software Engineering. Nr. 4, Jg. 5, 2000, S. 357-390.

Edwards, Newing /Application Integration/

Peter Edwards; Rod Newing: Application Integration for E-Business. Strategies for Integrating Key Applications, Systems and Processes. London 2000.

Eicker, Jung /Reengineering-Projekte/

Stefan Eicker; Reinhard Jung: Wirtschaftlichkeit von Reengineering-Projekten. In: Leonhard von Dobschütz; Josef Kistin; Eugen Schmidt (Hrsg.): IV-Controlling in der Praxis. Kosten und Nutzen der Informationsverarbeitung. Wiesbaden 1994, S. 113-131.

Eicker, Schnieder /Integrationsorientiertes Reengineering/

Stefan Eicker; Thomas Schnieder: Integrationsorientiertes Reengineering von Cobol-Altsystemen. In: Praxis der Informationsverarbeitung und Kommunikation (PIK). Nr. 3, Jg. 16, 1993, S. 162-168.

Eicker u. a. /Integrationsorientiertes Reengineering/

Stefan Eicker; Karl Kurbel; Wolfram Pietsch; Claus Rautenstrauch: Einbindung von Software-Altlasten durch integrationsorientiertes Reengineering. In: Wirtschaftsinformatik. Nr. 2, Jg. 34, 1992, S. 137-145.

El Emam, Madhavji /Measuring the Success/

Khaled El Emam; Nazim H. Madhavji: An Instrument for Measuring the Success of the Requirements Engineering Process in Information Systems Development. In: Empirical Software Engineering. Nr. 3, Jg. 1, 1996, S. 201-240.

Elmasri, Navathe /Database Systems/

Ramez Elmasri; Shamkant B. Navathe: Fundamentals of Database Systems. 4. Auflage, Boston u. a. 2004.

Falkenberg /Datenstruktur/

E. Falkenberg: Datenstruktur, konzeptionelle. In: Hans-Jochen Schneider (Hrsg.): Lexikon der Informatik und Datenverarbeitung. 4. Auflage, München - Wien 1998, S. 212.

Ferstl, Sinz /Wirtschaftsinformatik/

Otto K. Ferstl; Elmar J. Sinz: Grundlagen der Wirtschaftsinformatik, Band 1. 4. überarbeitete und erweiterte Auflage, München, Wien 2001.

Fincham /Application Integration/

Martin Fincham: Application integration: to invade. or not to invade?. In: Financial Middlewarespectra. Nr. 2, Jg. 13, May, 1999, S. 22-27.

Flick /Qualitative Forschung/

Uwe Flick: Qualitative Forschung. Theorie, Methoden, Anwendung in Psychologie und Sozialwissenschaften. 5. Auflage, Reinbeck 2000.

Francalanci /Implementation Effort/

Chiara Francalanci: Predicting the implementation effort of ERP projects: empirical evidence on SAP/R3. In: Journal of Information Technology. Nr. 1, Jg. 16, 2001, S. 33-48.

Frese /Organisation/

Erich Frese: Grundlagen der Organisation: Konzept - Prinzipien - Strukturen. 7. Auflage, Wiesbaden 1998.

Gerhardt /Softwareindustrie/

Gerhardt; Tilman: Strategie und Struktur in der deutschen Softwareindustrie. Eine industrieökonomische Untersuchung der Unternehmensentwicklung in der Softwarebranche. München 1992.

Gierharke /Geschäftsprozessmanagement/

Olaf Gierharke: Integriertes Geschäftsprozessmanagement. Effektive Organisationsgestaltung mit Workflow-, Workgroup- und Dokumentenmanagement-Systemen. 2. Auflage, Braunschweig, Wiesbaden 1998.

Gilb /Software Engineering Management/

Tom Gilb: Principles of Software Engineering Management. Wokingham, England u. a. 1988.

Gilpin /EAI Solution/

Mike Gilpin: How to Select an Enterprise Application Integration Solution. Giga Information Group. Cambridge, MA, 1999.

Goodyear /Enterprise System Architectures/

Mark u. a. Goodyear: Enterprise system architectures: building client/server and Web-based systems. Boca Raton, London, New York, Washington D. C. 2000.

Gorton, Thurman, Thomson /Application Integration/

Ian Gorton; Dave Thurman; Judi Thomson: Next Generation Application Integration: Challenges and New Approaches. In: Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03), November 03-06. Dallas, Texas 2003.

Grupp /Dokumentieren/

Bruno Grupp: EDV-Projekte richtig dokumentieren: Dokumentationstechniken, Dokumentationsrichtlinien, Dokumentationssoftware. Köln 1991.

Hansen, Neumann /Wirtschaftsinformatik/

Hans Robert Hansen; Gustaf Neumann: Wirtschaftsinformatik 2. Informationstechnik. 9. Auflage, Stuttgart 2005.

Hasselbring /Integration/

Wilhelm Hasselbring: Information System Integration. In: Communications of the ACM. Nr. 6, Jg. 43, 2000, S. 33-38.

Haupt /Dokumentation/

D. Haupt: Dokumentation. In: Hans Jochen Schneider (Hrsg.): Lexikon der Informatik und Datenverarbeitung. 4. Auflage, München - Wien 1997, S. 252.

Heilmann /Integration/

Heidi Heilmann: Integration: Ein zentraler Begriff der Wirtschaftsinformatik im Wandel der Zeit. In: Theorie und Praxis der Wirtschaftsinformatik (HMD). Nr. 150, Jg. 26, 1989, S. 46-58.

Heinrich, Heinzl, Roithmayr /Wirtschaftsinformatik-Lexikon/

Lutz J. Heinrich; Armin Heinzl; Friedrich Roithmayr: Wirtschaftsinformatik-Lexikon. 7. Auflage, München u. a. 2004.

Heyliger /Coupling/

George E. Heyliger: Coupling. In: John J. Marciniak (Hrsg.): Encyclopedia of Software Engineering: Volume 1. New York u. a. 1994, S. 220-228.

Hill, Laney /data integration/

Janelle Hill; Doug (METAGroup) Laney: The Future of Data Integration Technologies: a METAGroup White Paper sponsored by Synopsis, 2004.

Holl /Wirtschaftsinformatik/

Alfred Holl: Empirische Wirtschaftsinformatik und Erkenntnistheorie. In: Jörg Becker; Wolfgang König; Reinhard Schütte; Oliver Wendt; Stephan Zelewski (Hrsg.): Wirtschaftsinformatik und Wissenschaftstheorie: Bestandsaufnahme und Perspektiven. Wiesbaden 1999, S. 163-207.

Holten /Integration/

Roland Holten: Integration von Informationssystemen. In: Wirtschaftsinformatik. Nr. 1, Jg. 45, 2003, S. 41-52.

Humphrey, Singpurwalla /Predicting Software Productivity/

Watts S. Humphrey; Nozer D. Singpurwalla: Predicting (individual) Software Productivity. In: IEEE Transactions On Software Engineering. Nr. 2, Jg. 17, 1991, S. 196-207.

IFPUG /Counting Practices Manual/

International Function Point Users Group (IFPUG): Counting Practices Manual, Release 4.1. Westerville, OH 1999.

Johannesson, Perjons /Process Modelling/

Paul Johannesson; Erik Perjons: Design principles for process modelling in enterprise application integration. The Proceedings of the 12th international conference on advanced information systems engineering (CAiSE 00). In: Information Systems. Nr. 3, Jg. 26, 2001, S. 165-184.

Jones /Software Costs/

T. Capers Jones: Estimating software costs. New York u. a. 1998.

Jorgensen /Expert Estimation/

Magne Jorgensen: A Review of Studies on Expert Estimation of Software Development Effort. In: Journal of Systems and Software. Nr. 1-2, Jg. 70, 2004, S. 37-60.

Jorgensen, Molokken /Checklist/

Magne Jorgensen, Kjetil Molokken: A Preliminary Checklist for Software Cost Management. In: IEEE (Hrsg.): Proceedings of the Third International Conference On Quality Software (QSIC'03) Dallas, Texas, 6.-7. November 2003, S.134-140.

Jung /Integrationsorientiertes Reengineering/

Reinhard Jung: Wirtschaftlichkeitsfaktoren beim integrationsorientierten Reengineering. Verteilungsarchitektur und Integrationsschritte aus ökonomischer Sicht. Arbeitsbericht Nr. 16. Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik. Münster 1993.

Jung u. a. /Reengineering-Projekte/

Reinhard Jung; Andreas Rimpler; Thomas Schnieder; Alexander Teubner: Ein Experiment zur Untersuchung ausgewählter Kosteneinflussfaktoren bei Reengineering-Projekten. In: Wirtschaftsinformatik. Nr. 2, Jg. 38, 1996, S. 181-190.

Kaib /EAI/

Michael Kaib: Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Wiesbaden 2002.

Kazman, Bass /Quality Attributes/

Rick Kazman; Len Bass: Toward Deriving Software Architectures From Quality Attributes, Technical Report CMU/SEI-94-TR-10, Software Engineering Institute (SEI), Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania 1994.

Keller /EAI/

Wolfgang Keller: Enterprise Application Integration: Erfahrungen aus der Praxis. Heidelberg 2002.

Kirchmer /Standardsoftware/

Mathias Kirchmer: Geschäftsprozessorientierte Einführung von Standardsoftware: Vorgehen zur Realisierung strategischer Ziele. Wiesbaden 1996.

Kitchenham /Predictions/

Barbara Kitchenham: Making Process Predictions. In: Norman E. Fenton (Hrsg.): Software Metrics: A rigorous approach. London u. a. 1991, S. 132-149.

Kitchenham, Pfleeger, Fenton /Measurement Validation/

Barbara Kitchenham; Shari Lawrence Pfleeger; Norman Fenton: Towards a Framework for Software Measurement Validation. In: IEEE Transactions On Software Engineering. Nr. 12, Jg. 21, 1995, S. 929-944.

Kitchenham u. a. /Empirical Research/

Barbara A. Kitchenham; Shari Lawrence Pfleeger; Lesley M. Pickard; Peter W. Jones; David C. Hoaglin; Khaled El Emam; Jarett Rosenberg: Preliminary Guidelines for Empirical Research in Software Engineering. In: IEEE Transactions On Software Engineering. Nr. 8, Jg. 28, 2002, S. 721-734.

Klesse, Wortmann, Schelp /Applikationsintegration/

Mario Klesse; Felix Wortmann; Joachim Schelp: Erfolgsfaktoren der Applikationsintegration. In: Wirtschaftsinformatik. Nr. 4, Jg. 47, 2005, S. 259-267.

Krcmar /Informationsmanagement/

Helmut Krcmar: Informationsmanagement. 3. Auflage, Berlin u. a. 2003.

Kromer, Stucky /Mergers & Aquisitions/

Gerald Kromer; Wolffried Stucky: Die Integration von Informationsverarbeitungsressourcen im Rahmen von Mergers & Aquisitions. In: Wirtschaftsinformatik. Nr. 6, Jg. 44, 2002, S. 523-533.

Kubicek /Heuristische Bezugsrahmen/

Herbert Kubicek: Heuristische Bezugsrahmen und heuristisch angelegte Forschungsdesigns als Elemente einer Konstruktionsstrategie empirischer Forschung. In: Richard Köhler (Hrsg.): Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre: Bericht über die Tagung in Aachen, März 1976, Kommission Wissenschaftstheorie im Verband der Hochschullehrer für Betriebswirtschaft e. V. Stuttgart 1977, S. 3-36.

Kurbel /Software-Engineering/

Karl Kurbel: Stellungnahme zum Beitrag 'Braucht die Wirtschaftsinformatik ein eigenständiges Curriculum für Software-Engineering?'. In: Wirtschaftsinformatik. Nr. 6, Jg. 46, 2004, S. 495-496.

Kurbel, Rautenstrauch /Integration Engineering/

Karl Kurbel; Claus Rautenstrauch: Integration Engineering: Konkurrenz oder Komplement zum Information Engineering? - Methodische Ansätze zur Integration von Informationssystemen. In: Heidi Heilmann; Lutz Heinrich; Friedrich Roithmayr (Hrsg.): Information Engineering: Wirtschaftsinformatik im Schnittpunkt von Wirtschafts-, Sozial- und Ingenieurwissenschaften. München, Wien 1996, S. 167-191.

Lam /Enterprise Integration/

Wing Lam: Technical Risk Management on Enterprise Integration Projects. In: Communications of the Association for Information Systems. Nr. 20, Jg. 13, 2004, S. 291-316.

Lam /Success Factors/

Wing Lam: Investigating success factors in enterprise application integration: a case-driven analysis. In: European Journal of Information Systems. Nr. 2, Jg. 14, 2005, S. 175-187.

Lam, Shankararaman /Enterprise Integration/

Wing Lam; Venky Shankararaman: An Enterprise Integration Methodology. In: IT-Professional. Nr. 2, Jg. 6, 2004, S. 40-48.

Lane /System-of-System Integration/

Jo Ann Lane: Constructive Cost Model for System-of-System Integration. In: ACM-IEEE International Symposium on Empirical Software Engineering (ISESE), August 2004. 2004.

Lederer u. a. /Cost Estimating/
 Albert L. Lederer; Rajesh Mirani; Boon Slong Neo; Carol Pollard; Jayesh Prasad; K. Ramamurthy: Information System Cost Estimating: A Management Perspective. In: MIS Quarterly, Nr. 2, Jg. 14, 1990, S. 159-176.

Lee, Siau, Hong /Enterprise Integration/

Jinyoul Lee; Keng Siau; Soongoo Hong: Enterprise Integration with ERP and EAI. In: Communications of the ACM. Nr. 2, Jg. 46, 2003, S. 54-60.

Lehner /Wirtschaftsinformatik/

Franz Lehner: Wirtschaftsinformatik, Forschungsgegenstände und Erkenntnisverfahren. In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 505-507.

Ließmann /EAI/

Harald Ließmann: Enterprise Application Integration (EAI). In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 180-181.

Ließmann /Schnittstellenorientierung/

Harald Ließmann: Schnittstellenorientierung und Middleware-basierte Busarchitekturen als Hilfsmittel zur Integration heterogener betrieblicher Anwendungssysteme. Inaugural-Dissertation zur Erlangung des akademischen Grades eines Doktors der Wirtschafts- und Sozialwissenschaften. Wirtschafts- und Sozialwissenschaften, Friedrich-Alexander-Universität Erlangen-Nürnberg, Nürnberg 2000.

Ließmann, Wetzke /BPEL und BPMN/

Harald Ließmann; Jan Wetzke: Entwicklung und Management einer Serviceorientierten Integrationsinfrastruktur mithilfe von BPEL und BPMN. In: Stephan Aier; Marten Schönherr (Hrsg.): Enterprise Application Integration. Serviceorientierung und nachhaltige Architekturen. Berlin 2004, S. 197-228.

Linß /Nutzeffekte/

Heinz Linß: Integrationsabhängige Nutzeffekte der Informationsverarbeitung: Vorgehensmodell und empirische Ergebnisse. Wiesbaden 1995.

Linthicum /Adapters/

David S. Linthicum: The Evolution of Adapters. In: EAI Journal, Nr. 12, 2002, S. 37-40.

Linthicum /Application Integration/

David S. Linthicum: Next generation application integration: from simple information to Web services. Boston 2004.

Linthicum /B2B Integration/

David S. Linthicum: B2B Application Integration: e-Business-Enable Your Enterprise. Boston u. a. 2001.

Linthicum /EAI/

David S. Linthicum: Enterprise Application Integration. Boston u. a. 2000.

Losavio, Ortega, Pérez /EAI Quality/

Francisca Losavio; Dinarle Ortega; María Pérez: Towards a Standard EAI Quality Terminology. In: IEEE Computer Society (Hrsg.): Proceedings of the XXIII International Conference of the Chilean Computer Science Society (SCCC'03), 6-7 November 2003, Chillán, Chile. Los Alamitos, California u. a. 2003, S.119-129.

Losavio, Ortega, Pérez /EAI/

Fancisca Losavio; Dinarle Ortega; María Pérez: Modeling EAI. In: IEEE press (Hrsg.): Proceedings of the XII International Conference of the Chilean Computer Science Society (SCCC'02), 6-8 November 2002, Copiapó, Atacama, Chile, S.195-203.

Lublinsky /EAI Implementation/

Boris Lublinsky: Achieving the Ultimate EAI Implementation. In: EAI Journal, Nr. 2, 2001, S. S. 26-31.

Lublinsky, Farrell /Why EAI Fails/

Boris Lublinsky; Michael Jr. Farrell: Top 10 Reasons Why EAI Fails. In: EAI Journal, Nr. 12, 2002, S. 41-42.

Lünendonk (Hrsg.) /Business Innovation and Transformation/

Lünendonk (Hrsg.): Führende Business Innovation/Transformation Partner (BITP) in Deutschland 2004. <http://www.luenendonk.de/listen.php>, Aufruf am 04.02.2006.

Lünendonk (Hrsg) /IT-Beratung und Systemintegration/

Lünendonk (Hrsg.): TOP 25 IT-Beratungs- und Systemintegrations-Unternehmen in Deutschland 2004. <http://www.luenendonk.de/listen.php>, Aufruf am 04.02.2006.

Lünendonk (Hrsg.) /IT-Services/

Lünendonk (Hrsg.): Führende IT-Service Unternehmen in Deutschland 2004. <http://www.luenendonk.de/listen.php>, Aufruf am 04.02.2006.

Lutz /Patterns/

Jeffrey C. Lutz: EAI Architecture Patterns. In: EAI Journal, Nr. 3, 2000, S. 64-73.

Madapusi, D'Souza /ERP/

Arun Madapusi; Derrick D'Souza: Aligning ERP Systems with international strategies. In: Information Systems Management. Nr. 1, Jg. 22, 2005, S. 7-17.

McKeen, Smith /Enterprise Application Integration/

James D. McKeen; Heather A. Smith: New Developments in Practice II: Enterprise Application Integration. In: Communications of the Association for Information Systems, Nr. 31, Jg. 8, 2002, S. 451-466.

Mehta, Medvidovic, Phadke /Software Connectors/

Nikunj R. Mehta; Nenad Medvidovic; Sandeep Phadke: Towards a taxonomy of software connectors. In: International Conference on Software Engineering. Proceedings of the 22nd International Conference on Software Engineering. Limerick, Ireland 4-11 Juni 2000, S.178-187.

Mellis /Projektmanagement/

Werner Mellis: Projektmanagement der SW-Entwicklung. Eine umfassende und fundierte Einführung. Wiesbaden 2004.

Mellis /Systemdokumentation/

Werner Mellis: Systemdokumentation. In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 454-455.

Mertens /ERP/

Peter Mertens: Enterprise Resource Planning (ERP). In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 183.

Mertens /Integrierte Informationsverarbeitung/

Peter Mertens: Integrierte Informationsverarbeitung 1: Operative Systeme in der Industrie. 13. überarbeitete Auflage, Wiesbaden 2001.

Mertens, Holzner /Integrationsansätze/

Peter Mertens; Jochen Holzner: Eine Gegenüberstellung von Integrationsansätzen der Wirtschaftsinformatik. In: Wirtschaftsinformatik. Nr. 1, Jg. 34, 1992, S. 5-25.

META Group (Hrsg.) /META Trends 2004-05/

META Group (Hrsg.) : META Trends 2004/05. Integration & Development Strategies. Stamford 2004.

Midderhoff /V-Modell/

Rainer Midderhoff: Einführung in das V-Modell Version '97. In: Wolfgang Dröschel; Walter Heuser; Rainer Midderhoff (Hrsg.): Inkrementelle und objektorientierte Vorgehensweisen mit dem V-Modell 97. München, Wien 1998, S. 57-102.

Milling /Komplexe Systeme/

Peter Milling (Hrsg.): Entscheiden in komplexen Systemen, Wissenschaftliche Jahrestagung der Gesellschaft für Wirtschafts- und Sozialkybernetik vom 29. und 30. September 2000 in Mannheim. Berlin 2002.

Mische /Systems Integration/

Michael A. Mische: A Bird's-Eye View of Systems Integration. In: Information Strategy: The Executive's Journal. Nr. 4, Jg. 13, 1997, S. 33-39.

Morris u. a. /Database/

K. C. Morris; Mary Mitchell; Christopher Dabrowski; Elizabeth Fong: Database Management Systems in Engineering. In: John J. Marciniak (Hrsg.): Encyclopedia of Software Engineering: Volume 1. New York u. a. 1994, S. 282-308.

Müller-Böhling /Organisationsforschung/

Müller-Böhling: Organisationsforschung, Methodik der empirischen. In: Erich Frese (Hrsg.): Handwörterbuch der Organisation. 3. Auflage, Stuttgart 1992, S. 1491-1505.

Noth, Kretzschmar /Aufwandschätzung/

Thomas Noth; Mathias Kretzschmar: Aufwandschätzung von DV-Projekten. Darstellung und Praxisvergleich der wichtigsten Verfahren. 2. Auflage, Berlin u. a. 1986.

Österle /Integration/

Hubert Österle: Integration: Schlüssel zur Informationsgesellschaft. In: Hubert Österle; Rainer Riehm; Petra Vogler (Hrsg.): Middleware: Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten. Braunschweig/Wiesbaden 1996, S. 1-23.

Park /Checklist/

Robert E. Park: A Manager's Checklist for Validating Software cost and Schedule Estimates. Special Report, CMU/SEI-95-SR-004, January 1995. Carnegie Mellon University, Software Engineering Institute. Pittsburgh, Pennsylvania, 1995.

Pillai, Nair /Estimation/

Krishnakumar Pillai; Sukumaran Nair: A Model for Software Development Effort and Cost Estimation. In: IEEE Transactions On Software Engineering. Nr. 8, Jg. 23, 1997, S. 485-497.

Pinkston /Integration/

Jeff Pinkston: The Ins and Outs of Integration: How EAI Differs from B2B Integration. In: EAI Journal, Nr. 8 (August), 2001, S. 48-52.

Premkumar, Ramamurthy, Saunders /Information Processing/

G. Premkumar; K. Ramamurthy; Carol Stauk Saunders: Information Processing View of Organizations: An Exploratory Examination of the Fit in the Context of Interorganizational Relationships. In: Journal of Management Information Systems. Nr. 1, Jg. 22, 2005, S. 257-294.

Putz-Osterloh /Entscheidungsverhalten/

Wiebke Putz-Osterloh: Entscheidungsverhalten. In: Erich Frese (Hrsg.): Handwörterbuch der Organisation. 3., völlig neu gestaltete Auflage, Stuttgart 1992, S. 585-599.

Quack /CIO/

Karin Quack: Zwölf Themen, die den CIO bewegen. In: Computerwoche Spezial. 10. Dez. 2004, S. 16-17.

Quack /IT-Trends 2002/

Karin Quack: IT-Trends 2002: Zurück zur Vernunft. In: Computerwoche. Nr. 11, Jg. 29, 2002, S. 37.

Rask, Laamanen, Lyytinen /Function Point and Function Bang/

Raimo Rask; Petteri Laamanen; Kalle Lyytinen: Simulation and Comparison of Albrecht's Function Point and DeMarco's Function Bang Metrics in a CASE Environment. In: IEEE Transactions On Software Engineering. Nr. 7, Jg. 19, 1993, S. 661-671.

Reifer, Boehm, Gangadharan /Cost of Security/

Donald J. Reifer; Barry W. Boehm; Murali Gangadharan: Estimating the Cost of Security for COTS Software. In: H. Erdogmus; T. Weng (Hrsg.): In Proceedings of 2nd International Conference on COTS-Based Software Systems, February 2003. Berlin, Heidelberg 2003, S.178–186.

Reinhard Jung: Wirtschaftlichkeitsfaktoren beim integrationsorientierten Reengineering. Verteilungsarchitektur und Integrationsschritte aus ökonomischer Sicht. Arbeitsbericht Nr. 16. Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik. Münster, 1993.

Renk /Return-on-Investment/

Thomas Renk: Auf der Suche nach dem Return-on-Investment. In: Computerwoche. Nr. 36, Jg. 28, 2001, S. 62-63.

Riehm, Vogler /Middleware/

Rainer Riehm; Petra Vogler: Middleware: Infrastruktur für die Integration. In: Hubert Österle; Rainer Riehm; Petra Vogler (Hrsg.): Middleware: Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten. Wiesbaden 1996, S. 25-135.

Rine, Nada, Jaber /Adapters/

David Rine; Nader Nada; Khaled Jaber: Using adapters to reduce interaction complexity in reusable component-based software development. In: Symposium on Software Reusability. Proceedings of the 1999 symposium on Software reusability. Los Angeles 1999, S.37-43.

Robey, Markus /Rituals/

D. Robey; M. L. Markus: Rituals in Information System Design. In: MIS Quarterly, Jg. 8, Nr. 1, 1984, S. 5-15.

Roch /Methodology for EAI - Part I/

Eric Roch: A Software Development Methodology for EAI. Part I. In: Business Integration Journal, Nr. 9, 2004, S. 23-26.

Roch /Methodology for EAI - Part II/

Eric Roch: A Software Development Methodology for EAI. Part II. In: Business Integration Journal, Nr. 10, 2004, S. 14-16.

Rossiter /Scale Development/

John R. Rossiter: The C-OAR-SE procedure for scale development in marketing. In: International Journal of Research in Marketing. Nr. 4, Jg. 19, 2002, S. 305-335.

Ruh, Maginnis, Brown /EAI/

William A. Ruh; Francis X. Maginnis; William J. Brown: Enterprise Application Integration. A Wiley Tech Brief. New York u. a. 2001.

Rupietta /Benutzerdokumentation/

Walter Rupietta: Benutzerdokumentation für Softwareprodukte. Mannheim - Wien - Zürich 1987.

Sage, Cuppan /Systems of Systems/

Andrew P. Sage; Christopher D. Cuppan: On the Systems Engineering and Management of Systems of Systems and Federations of Systems. In: Information Knowledge Systems Management. Nr. 4, Jg. 2, 2001, S. 325-345.

Sandoe, Corbitt, Boykin /Enterprise Integration/

Kent Sandoe; Gail Corbitt; Raymond Boykin: Enterprise Integration. New York u. a. 2001.

Scheer /ARIS/

August-Wilhelm Scheer: ARIS - Vom Geschäftsprozess zum Anwendungssystem. 4., durchgesehene Auflage, Berlin u. a. 2002.

Schierenbeck /Betriebswirtschaftslehre/

Henner Schierenbeck: Grundzüge der Betriebswirtschaftslehre. 16., vollständig überarbeitete und erweiterte Auflage, München, Wien 2003.

Schmietendorf u. a. /EAI/

Andreas Schmietendorf; Evgeni Dimitrov; Jens Lezius; Reiner Dumke: Enterprise Application Integration - Reifegrad, Architektur und Vorgehensweisen. In: Stefan Meinhardt; Karl Popp (Hrsg.): Enterprise Portale & Enterprise Application Integration. HMD 225. Heidelberg 2002, S. 114-123.

Schumann /Großintegrierte Informationsverarbeitung/

Matthias Schumann: Betriebliche Nutzeffekte und Strategiebeiträge der großintegrierten Informationsverarbeitung. Berlin u. a. 1992.

Schwinn, Winter /Enterprise Application Integration/

Alexander Schwinn; Robert Winter: Success Factors and Performance Indicators for Enterprise Application Integration. In: Romano, N. C. Jr. (Hrsg.): Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE USA August 11th - 14th 2005. 2005 Atlanta (GA), S.2179-2189.

Searle /Speech Acts/

John R. Searle: Speech acts. An essay in the philosophy of language. London 1969.

Seibt /Anwender/

Dietrich Seibt: Anwender. In: Hans-Jochen Schneider (Hrsg.): Lexikon der Informatik und Datenverarbeitung. München, Wien 1998, S. 43.

Seibt /Anwendungssystem/

Dietrich Seibt: Anwendungssystem. In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 46-47.

Seibt /Projektaufwandschätzung/

Dietrich Seibt: Probleme der Projektaufwandschätzung aus betriebswirtschaftlicher Sicht. In: A. Büllsbach; Thomas Heymann (Hrsg.): Informationsrecht 2000 - Perspektiven für das nächste Jahrzehnt. Köln 2000, S. 29-50.

Seibt /Softwaresysteme/

Dietrich Seibt: Organisation von Softwaresystemen. Wiesbaden 1972.

Seibt /Systemsoftware/

Dietrich Seibt: Systemsoftware. In: Hans-Jochen Schneider (Hrsg.): Lexikon der Informatik und Datenverarbeitung. 4. Auflage, München, Wien 1998, S. 858.

Seibt /Vorgehensmodell/

Dietrich Seibt: Vorgehensmodell. In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a. 2001, S. 498-500.

Singletary /Applications Integration/

Less Singletary: Empirical Study of Attributes and Perceived Benefits of Applications Integration for Enterprise Systems. Doctoral dissertation. Louisiana State University - E. J. Ourso College of Business Administration, Baton Rouge, LA 2003.

Sinz, Mertens /Kopplung/

Elmar J. Sinz; Peter Mertens: Kopplung von Anwendungssystemen: WI - Editorial zum Schwerpunktthema. In: Wirtschaftsinformatik. Nr. 5, Jg. 44, 2002, S. 409-410.

Smith u. a. /Enterprise Integration/

Dennis Smith; Liam O'Brian; Mario Barbacci; Francois Coallier: A Roadmap for Enterprise Integration. In: Proceedings of the 10th International Workshop on Software Technology and Engineering Practice (STEP'02). 2002.

Stadtler /Supply Chain Management/

Hartmut Stadtler: Supply Chain Management. An Overview. In: Hartmut Stadtler; Christoph Kilger (Hrsg.): Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies. 3. Auflage, Berlin, Heidelberg 2005, S. 9-35.

Stahlknecht, Hasenkamp /Wirtschaftsinformatik/

Peter Stahlknecht; Ulrich Hasenkamp: Einführung in die Wirtschaftsinformatik. 9. Auflage, Berlin, Heidelberg 1999.

Steinke /Qualitative Research/

Ines Steinke: Quality Criteria in Qualitative Research. In: Uwe Flick; Ernst von Kardorff; Ines Steinke (Hrsg.): A Companion to Qualitative Research. London, Thousand Oaks, New Delhi 2004, S. 184-190.

Stelzer /Software-Qualitätsmanagement/

Dirk Stelzer: Möglichkeiten und Grenzen des prozessorientierten Software-Qualitätsmanagements. Köln 1998.

Stevenson /Software Engineering Productivity/

C. Stevenson: Software engineering productivity. A practical guide. London 1995.

Tan /Business-IT Alignment/

Felix B. Tan: Exploring Business-IT Alignment Using the Repertory Grid. In: Proceedings of the 10th Australasian Conference on Information Systems, 1.-3. December 1999, Wellington, New Zealand 1999, S.931-943.

Themistocleous, Irani /Evaluating Application Integration/

Marinos Themistocleous; Zahir Irani: Evaluating Application Integration: An Exploratory Case Study. In: Strong, D.; Straub, D. (Hrsg.): Seventh Americas Conference on Information Systems. 3.-5. August 2001, Boston, Massachusetts 2001, S.1376-1380.

Themistocleous, Irani /Taxonomy for Application Integration/

Marinos Themistocleous; Zahir Irani: Novel taxonomy for application integration. In: Benchmarking: An International Journal. Nr. 2, Jg. 9, 2002, S. 154-165.

Thompson u. a. /Application Integration Suites 2Q04/

J. Thompson; R. Schulte; M. Cantara; J. Correia; K. Iijima; F. Kenney; B. Lheureux; Y. Natis; M. Pezzini; J. Sinur: Magic Quadrant for Application Integration Suites, 2Q04. Gartner Research Note. Gartner Group Inc., 14. Mai 2004.

Trapp, Otto /EAI/

Ralph C. Trapp; Alfred Otto: Einsatzmöglichkeiten von EAI bei Mergers & Acquisitions. In: Handbuch der modernen Datenverarbeitung (HMD), Nr. 225, 2002, S. 102-113.

Trittmann u. a. /Projektgestaltung/

Ralph Trittmann; Werner Mellis; Holger Wagner; Rasmus Bergmann; Oral Avci: Sieg der Moderne über die Tradition? Ergebnisse einer empirischen Untersuchung zur Projektgestaltung in der Softwareentwicklung. In: Projektmanagement aktuell. Nr. 4, Jg. 16, 2005, S. 12-17.

Vasconcelos u. a. /Architectural Framework/

André Vasconcelos; Miguel Mira da Silva; António Fernandes; José Tribolet: An Information System Architectural Framework for Enterprise Application Integration. In: IEEE Computer Society (Hrsg.): Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Jan. 2004, Hilton Waikoloa Village Big Island, Hawaii. Hawaii 2004.

Voigtmann, Zeller /Integrationssysteme/

Peter Voigtmann; Thomas Zeller: Enterprise Application Integration und B2B Integration im Kontext von Electronic Business und Elektronischen Marktplätzen. Teil II: Integrationssysteme und Fallbeispiele. FORWIN-Bericht-Nr.: FWN-2003-001. Universität Erlangen-Nürnberg, Lehrstuhl Prof. Dr. Dr. h. c. mult. Peter Mertens, Bayrischer Forschungsverbund Wirtschaftsinformatik. Nürnberg, 2003.

Walston, Felix /Programmin Measurement And Estimation/

C. E. Walston; C. P. Felix: A method of programming measurement and estimation. In: IBM Systems Journal. Nr. 1, Jg. 16, 1977, S. 54-73.

Wangler, Paheerathan /Integration of Organizational IT Systems/

B. Wangler; S. J. Paheerathan: Horizontal and Vertical Integration of Organizational IT Systems. In: S. Brinkkemper; E. Lindencrona; A. Solvberg (Hrsg.): Information Systems Engineering: State of the the Art and Research. Heidelberg 2000, S. 79-90.

Wedekind /Datenbanksystem/

Hartmut Wedekind: Datenbanksystem. In: Peter Mertens; Andrea Back; Jörg Becker; Wolfgang König; Hermann Krallmann; Bodo Rieger; August-Wilhelm Scheer; Dietrich Seibt; Peter Stahlknecht; Horst Strunz; Rainer Thome; Hartmut Wedekind (Hrsg.): Lexikon der Wirtschaftsinformatik. 4. Auflage, Berlin u. a., S. 139-140.

Wiederhold, Quian /Database/

Gio Wiederhold; Xiaolei Quian: Database Engineering. In: John J. Marciniak (Hrsg.): Encyclopedia of Software Engineering: Volume 1. New York u. a. 1994, S. 269-282.

Witte /Entscheidungsprozesse/

Eberhard Witte: Entscheidungsprozesse. In: Erich Frese (Hrsg.): Handwörterbuch der Organisation. 3., völlig neu gestaltete Auflage, Stuttgart 1992, S. 552-565.

Wollnik /Erfahrungswissen/

Michael Wollnik: Die explorative Verwendung systematischen Erfahrungswissens: Plädoyer für einen aufgeklärten Empirismus in der Betriebswirtschaftslehre. In: Richard Köhler (Hrsg.): Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre: Bericht über die Tagung in Aachen, März 1976, Kommission Wissenschaftstheorie im Verband der Hochschullehrer für Betriebswirtschaft e. V. Stuttgart 1977, S. 37-64.

Wrazen /EAI/

Ed Wrazen: Enterprise Integration - Tools not the complete solution. In: Database & Network Journal. Nr. 5, Jg. 29, 1999, S. 3-5.

Yee, Apte /e-Business Enterprise/

Andre Yee; Atul Apte: Integrating Your e-Business Enterprise. Indianapolis, IN 2001.

Anhang A: Detailinformationen zu den erhobenen Konstrukten

Im Folgenden werden die einzelnen Konstrukte der erläuterten Gruppen gegliedert nach den Themenbereichen dargestellt.

Die gewählte tabellarische Darstellungsform basiert auf einem einheitlichen Aufbau. In der linken Spalte der Tabellen werden die Bezeichnungen der einzelnen Gruppen angegeben. In den anschließenden Spalten werden die zugeordneten Konstrukte aufgeführt. Die beiden polaren Ausprägungen der Konstrukte sind stets in gleicher Weise angeordnet: die Ausprägung, die gemäß der Ansicht der Untersuchungsteilnehmer mit einem geringeren Entwicklungsaufwand verbunden ist, ist in der linken der beiden Spalten aufgeführt; die Ausprägung, die gemäß der Ansicht der Untersuchungsteilnehmer mit einem höheren Entwicklungsaufwand verbunden ist, ist in der rechten der beiden Spalten aufgeführt. In der anschließenden Spalte ist die Häufigkeit der Konstrukte angegeben, d. h. die Anzahl der Untersuchungsteilnehmer, die das Konstrukt im Rahmen der Interviews genannt haben.

In den restlichen Spalten ist die Kommentierung der Konstrukte durch die Untersuchungsteilnehmer erfasst, d. h. welcher Anteil der Untersuchungsteilnehmer, die das betreffende Konstrukt genannt haben, es als charakteristisch für Integrationsvorhaben bzw. als von hoher (+), mittlerer (o) oder niedriger (-) Bedeutung für den Entwicklungsaufwand beurteilt haben.

Es muss jedoch beachtet werden, dass diese Urteile lediglich auf eine geringe Anzahl von Auskunftspersonen zurückzuführen sind. Wie erläutert wurden die Konstrukte ausschließlich in den Interviews kommentiert, in denen sie erhoben wurden. In vielen Fällen können daher nur die Einschätzungen einzelner Untersuchungsteilnehmer wiedergegeben werden. Zudem erfolgte die Kommentierung der einzelnen Konstrukte stets im spezifischen Kontext der weiteren Konstrukte, die in den betreffenden Interviews notiert worden waren. Die angegebenen Prozentwerte sollten daher ausschließlich im Sinne des verfolgten qualitativen und ideographischen Forschungsansatzes interpretiert werden: als Kommentierung der Konstrukte durch die Untersuchungsteilnehmer, als Ausdruck ihrer persönlichen Erfahrungswerte und Ansichten. Quantitative bzw. normative Schlüsse, insbesondere auf die Bedeutung der in den Konstrukten beschriebenen Merkmale für den Entwicklungsaufwand, sind dagegen nicht möglich.

Konstrukte zu den Anforderungen an die Integrationslösung

Dem Themenbereich der Anforderungen an die Integrationslösung wurden 15 Gruppen zugeordnet, die insgesamt 55 der erhobenen Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Der allgemeine Umfang der Aufgabenstellung	Der Integrationsumfang ist gering ("kleiner scope").	Der Integrationsumfang ist groß ("großer scope").	1	100 %	100 %	0 %	0 %
	Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht ist gering.	Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht sind groß.	4	50 %	50 %	50 %	0 %
Die Anzahl, die Art und die Komplexität der zu unterstützenden Geschäftsprozesse	Die Anzahl der zu unterstützenden Geschäftsprozesse ist gering.	Die Anzahl der zu unterstützenden Geschäftsprozesse ist groß.	2	50 %	50 %	50 %	0 %
	Es gilt Geschäftsprozesse einer Art ("Prozessfamilie") zu unterstützen.	Es gilt Geschäftsprozesse vieler unterschiedlicher Arten ("Prozessfamilien") zu unterstützen.	1	100 %	0 %	100 %	0 %
	Die Komplexität eines zu unterstützenden Geschäftsprozesses ist gering.	Die Komplexität eines zu unterstützenden Geschäftsprozesses ist groß.	1	100 %	0 %	100 %	0 %
Die Anzahl und die Komplexität der zu unterstützenden Use Cases	Die Anzahl der zu unterstützenden Use Cases ist gering.	Die Anzahl der zu unterstützenden Use Cases ist groß.	1	100 %	0 %	100 %	0 %
	Die Komplexität eines zu unterstützenden Use Cases ist gering.	Die Komplexität eines zu unterstützenden Use Cases ist groß.	1	100 %	0 %	100 %	0 %
Die mit der Anwendungsintegration verfolgte Perspektive	Die Anwendungsintegration betrifft ausschließlich technische Aspekte.	Die Anwendungsintegration betrifft technische und fachliche Aspekte.	1	100 %	100 %	0 %	0 %
	Die Anwendungssysteme werden integriert, um damit Benutzer zu bedienen. Die Ansprüche an die Anwendungsintegration werden somit durch Menschen gestellt.	Die Anwendungssysteme werden integriert, um damit weitere Softwaresysteme zu bedienen. Die Ansprüche an die Anwendungsintegration werden somit durch Softwaresysteme gestellt.	1	100 %	100 %	0 %	0 %
	Die Integrationslösung basiert auf der Daten-Ebene.	Die Integrationslösung umfasst auch Interaktionen mit Benutzern und macht damit insbesondere die Unterstützung von Berechtigungskonzepten und Workflows erforderlich.	1	100 %	0 %	0 %	100 %
	Die Anwendungsintegration ist informationsorientiert.	Die Anwendungsintegration ist prozessorientiert.	1	100 %	0 %	0 %	100 %
	Die Interaktionen der zu integrierenden Anwendungssysteme erfolgen vollständig automatisiert.	Während der Interaktionen der zu integrierenden Anwendungssysteme sind manuelle Interventionen erforderlich.	1	100 %	0 %	0 %	100 %
	Die zu unterstützenden Geschäftsprozesse werden nicht vollständig automatisiert.	Die zu unterstützenden Geschäftsprozesse werden vollständig automatisiert.	1	100 %	0 %	0 %	100 %
Die Neuartigkeit der Integrationslösung (Fortsetzung folgende Seite)	Die zu entwickelnde Integrationslösung baut auf einer bereits bestehenden Integrationslösung auf.	Die zu entwickelnde Integrationslösung wird vollständig neu aufgebaut.	3	67 %	100 %	0 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Neuartigkeit der Integrationslösung (Fortsetzung)	Eine bereits bestehende Integrationslösung soll nachgebaut und ersetzt werden.	Bislang ist keine Integrationslösung vorhanden.	1	0 %	100 %	0 %	0 %
	Die Integrationslösung basiert auf einer etablierten und verifizierten Architektur.	Die Integrationslösung basiert auf einer nicht ausreichend erprobten Architektur.	1	100 %	100 %	0 %	0 %
	Eine bereits zentralisierte Anwendungslandschaft soll durch eine zentrale Lösung integriert werden. Es muss somit eine bestehende IT-Architektur umstrukturiert werden.	Eine verteilte Anwendungslandschaft soll durch die Anwendungsintegration zentralisiert werden. Es muss somit grundsätzlich eine andere IT-Architektur geschaffen werden.	1	0 %	100 %	0 %	0 %
	Es ist nicht erforderlich, fachliche Funktionalität neu zu entwickeln.	Es ist erforderlich, fachliche Funktionalität neu zu entwickeln.	1	0 %	0 %	100 %	0 %
Die Neuartigkeit der durch die Integrationslösung zu unterstützenden Geschäftsprozesse	Die zu unterstützenden Geschäftsprozesse sind etabliert.	Die zu unterstützenden Geschäftsprozesse müssen erst definiert werden.	1	0 %	0 %	100 %	0 %
	Ein durch die Integrationslösung unterstützter Geschäftsprozess ist ein Ersatz für einen einheitlichen bestehenden Geschäftsprozess.	Ein durch die Integrationslösung unterstützter Geschäftsprozess ist ein Ersatz für mehrere unterschiedliche Geschäftsprozesse.	1	100 %	0 %	100 %	0 %
Die Klarheit der Anforderungen	Der Kunde kennt und versteht das Integrationsproblem.	Der Kunde muss erst eine Integrationslösung sehen und verstehen, um das Integrationsproblem genau zu verstehen.	1	0 %	100 %	0 %	0 %
	Es gibt eine klare Aufgabenstellung.	Der Kunde ist sich über seine Ziele unklar. Die Aufgabenstellung entwickelt sich nach und nach.	2	0 %	50 %	50 %	0 %
	Die zu unterstützenden Geschäftsprozesse sind klar definiert.	Die zu unterstützenden Geschäftsprozesse sind nicht klar definiert.	1	0 %	0 %	100 %	0 %
	Die fachlichen Interdependenzen zwischen den zu integrierenden Anwendungssystemen können genau definiert werden.	Die fachlichen Interdependenzen zwischen den zu integrierenden Anwendungssystemen können nicht definiert werden.	1	100 %	100 %	0 %	0 %
	Es erfolgt keine Experimentierung im Laufe des Integrationsvorhabens.	Es erfolgt viel Experimentierung im Laufe des Integrationsvorhabens.	1	0 %	0 %	100 %	0 %
	Die Anforderungen sind klar definiert und dokumentiert.	Die Anforderungen sind nicht klar definiert und dokumentiert.	4	0 %	75 %	25 %	0 %
	Die Dokumentation der Anforderungen basiert auf formaler Modellierung. Es gibt nur geringen Interpretationsspielraum bezüglich des Analyseergebnisses.	Die Dokumentation der Anforderungen ist lediglich verbalsprachlich. Das Analyseergebnis ist frei interpretierbar.	1	0 %	100 %	0 %	0 %
	Die Projektmitarbeiter wissen genau was sie machen müssen.	Die Projektmitarbeiter füllen Lächer in ihrem Verständnis mit eigenen Entscheidungen.	1	0 %	0 %	100 %	0 %
	Es gibt eine inhaltliche Interpretation von Qualität. Die Anforderungen an die Qualität der Integrationslösung sind gestaltbar.	Es gibt eine formale Interpretation von Qualität. Die Anforderungen an die Qualität der Integrationslösung sind nicht inhaltlich sondern quantitativ festgelegt.	1	100 %	0 %	100 %	0 %

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rak- teristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Änderungs- dynamik der Anfor- derungen	Die Anforderungen sind klar definiert und stabil.	Im Laufe des Integrationsvorhabens treten neue Anforderungen auf.	3	33 %	67 %	33 %	0 %
	Es gibt eine klare Aufgabenstellung.	Das Integrationsvorhaben unterliegt einem starken Einfluss von Außen. Es findet in einem dynamischen Umfeld statt.	1	100 %	100 %	0 %	0 %
	Das Integrationsvorhaben erfolgt in einem stabilen und bekannten Systemumfeld.	Das Integrationsvorhaben erfolgt in einem dynamischen Systemumfeld. Das Systemumfeld ändert sich im Lauf des Integrationsvorhabens.	1	100 %	0 %	100 %	0 %
	Der Projektumfang ändert sich während der Projektlaufzeit nicht.	Der Projektumfang ändert sich während der Projektlaufzeit.	2	0 %	50 %	50 %	0 %
	Die Organisation des Kunden ändert sich nicht im Laufe des Integrationsvorhabens.	Die Organisation des Kunden ändert sich im Laufe des Integrationsvorhabens.	1	0 %	0 %	0 %	100 %
Der Umgang mit Änderungen der Anforderungen	Es gibt ein etabliertes Change Request System.	Die Verwaltung der Change Requests muss neu erarbeitet und umgesetzt werden.	1	0 %	0 %	100 %	0 %
Die Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung	Die qualitätsbezogenen/nicht-funktionalen Anforderungen an die Integrationslösung sind gering.	Die qualitätsbezogenen/nicht-funktionalen Anforderungen an die Integrationslösung sind hoch.	2	50 %	100 %	0 %	0 %
	Die Sicherheitsanforderungen an die Integrationslösung sind gering.	Die Sicherheitsanforderungen an die Integrationslösung sind hoch.	4	50 %	25 %	75 %	0 %
	Es ist nicht kritisch, dass über die zu integrierenden Anwendungssysteme hinweg Transaktionssicherheit besteht.	Transaktionssicherheit über die zu integrierenden Anwendungssysteme hinweg ist erforderlich.	3	67 %	67 %	33 %	0 %
	Die Anforderungen an die Verfügbarkeit der Integrationslösung sind gering.	Die Anforderungen an die Verfügbarkeit der Integrationslösung sind hoch.	4	50 %	50 %	50 %	0 %
	Die Anforderungen an die Ausfallsicherheit der Integrationslösung sind gering.	Die Anforderungen an die Ausfallsicherheit der Integrationslösung sind hoch.	2	50 %	50 %	50 %	0 %
	Die Anforderungen an die Performanz der Integrationslösung sind gering.	Die Anforderungen an die Performanz der Integrationslösung sind hoch.	4	25 %	75 %	25 %	0 %
	Die Anforderungen an die Skalierbarkeit der Integrationslösung sind gering.	Die Anforderungen an die Skalierbarkeit der Integrationslösung sind hoch.	1	0 %	100 %	0 %	0 %
	Die Anforderungen an die Revisionsicherheit der Integrationslösung sind gering.	Die Anforderungen an die Revisionsicherheit der Integrationslösung sind hoch.	1	0 %	0 %	100 %	0 %
	Die Integrationslösung besitzt keine komplexen Fehlerbehandlungsmechanismen. Fehler werden nur protokolliert.	Die Integrationslösung besitzt komplexe Fehlerbehandlungsmechanismen.	2	0 %	0 %	100 %	0 %
	Bei der Gestaltung der Integrationslösung werden nur unmittelbare Ziele verfolgt.	Bei der Gestaltung der Integrationslösung werden eine hohe Flexibilität und Wiederverwendbarkeit angestrebt.	3	33 %	33 %	33 %	33 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Anforderungen an die Protokollierung der Interaktionen zwischen den zu integrierenden Anwendungssystemen	Bei der Interaktion von Anwendungssystemen besteht kein fachlicher Überwachungsbedarf ("track and trace").	Bei der Interaktion von Anwendungssystemen besteht großer fachlicher Überwachungsbedarf ("track and trace").	1	0 %	0 %	100 %	0 %
	Für die Interaktionen der zu integrierenden Anwendungssysteme ist lediglich eine eingeschränkte Kontakthistorie erforderlich.	Für die Interaktionen der zu integrierenden Anwendungssysteme ist eine umfangreiche Kontakthistorie erforderlich.	1	100 %	100 %	0 %	0 %
Die Anzahl der Benutzer und Benutzerrollen	Die Anzahl der Benutzer ist gering.	Die Anzahl der Benutzer ist groß.	1	0 %	0 %	0 %	100 %
	Die Anzahl der Benutzerrollen ist gering.	Die Anzahl der Benutzerrollen ist groß.	1	0 %	0 %	0 %	100 %
Die Verteilung der Nutzung der Integrationslösung	Die Integrationslösung wird zentral an einem Standort genutzt.	Die Integrationslösung wird dezentral an verschiedenen Standorten genutzt.	1	0 %	100 %	0 %	0 %
Die Notwendigkeit des Abgleichs der Stammdaten der zu integrierenden Anwendungssysteme	Es ist keine initiale Stammdatensynchronisation erforderlich.	Es ist eine initiale Stammdatensynchronisation erforderlich.	1	0 %	0 %	100 %	0 %
	Es ist keine Stammdatensynchronisation zur Laufzeit notwendig.	Es ist eine Stammdatensynchronisation zur Laufzeit erforderlich.	1	100 %	0 %	100 %	0 %
	Es gibt einen festen und durchgehenden Satz von Stammdaten.	Die Stammdaten weisen eine hohe Dynamik auf.	1	100 %	0 %	100 %	0 %
Die Notwendigkeit einer Migration	Es ist keine Migration von Daten erforderlich.	Es ist eine Migration von Daten erforderlich.	1	100 %	100 %	0 %	0 %
	Ein Parallelbetrieb zu migrierender Systeme ist nicht erforderlich.	Ein Parallelbetrieb zu migrierender Systeme ist erforderlich.	2	100 %	100 %	0 %	0 %

Konstrukte zu den Charakteristika der zu integrierenden Anwendungssysteme

Dem Themenbereich der Charakteristika der zu integrierenden Anwendungssysteme wurden 12 Gruppen zugeordnet, die insgesamt 35 der persönlichen Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rakte- ristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Anzahl und die Komplexität der zu integrierenden Anwendungs- systeme	Die Anzahl der zu integrierenden Anwendungssysteme ist gering.	Die Anzahl der zu integrierenden Anwendungssysteme ist groß.	6	100 %	67 %	33 %	0 %
	Die Komplexität eines zu integrierenden Anwendungssystems ist gering.	Die Komplexität eines zu integrierenden Anwendungssystems ist groß.	1	0 %	0 %	100 %	0 %
Die Arten zu integrierender Anwen- dungssysteme	Im Rahmen der Integrationslösung werden keine Telekommunikationssysteme angebunden.	Im Rahmen der Integrationslösung werden Telekommunikationssysteme angebunden.	1	100 %	100 %	0 %	0 %
	Ein zu integrierendes Anwendungssystem ist in räumlicher Hinsicht statisch.	Ein zu integrierendes Anwendungssystem ist in räumlicher Hinsicht mobil (z. B. im Zusammenhang mit Telemetrie).	1	0 %	100 %	0 %	0 %
	Eine Interaktion findet zwischen zwei Server-Systemen statt.	Eine Interaktion findet zwischen Server- und Client-Systemen statt.	1	100 %	0 %	100 %	0 %
	Alle zu integrierenden Anwendungssysteme sind Standardsoftwaresysteme.	Alle zu integrierenden Anwendungssysteme sind Individualsoftwaresysteme.	1	100 %	0 %	100 %	0 %
Die Verfügbarkeit der Schnittstellen der zu integrierenden Anwendungs- systeme	Ein zu integrierendes Anwendungssystem hat wohldefinierte Schnittstellen.	Ein zu integrierendes Anwendungssystem hat keine wohldefinierten Schnittstellen.	2	100 %	50 %	50 %	0 %
	Die Reife einer Schnittstelle eines Anwendungssystems ist hoch.	Eine Schnittstelle eines Anwendungssystems muss neu entwickelt werden.	1	100 %	0 %	100 %	0 %
	Der Kunde hat vollständige Nutzungsmöglichkeiten und -rechte der Schnittstellen eines zu integrierenden Anwendungssystems sowie der zugehörigen Dokumentation.	Der Kunde hat keine Nutzungsmöglichkeiten und -rechte der Schnittstellen eines zu integrierenden Anwendungssystems sowie der zugehörigen Dokumentation.	1	100 %	100 %	0 %	0 %
Die Eignung der Schnittstellen der zu integrierenden Anwendungs- systeme für die Integrationslösung	Ein zu integrierendes Anwendungssystem ist technisch für die Integration vorbereitet bzw. geeignet.	Ein zu integrierendes Anwendungssystem ist technisch nicht für die Integration vorbereitet bzw. geeignet.	2	100 %	50 %	50 %	0 %
	Ein zu integrierendes Anwendungssystem besitzt eine geeignete Schnittstelle für eine erforderliche Online-Verarbeitung.	Eine Online-Verarbeitung muss unter Verwendung einer Batch-Schnittstelle des zu integrierenden Anwendungssystems realisiert werden.	1	100 %	0 %	0 %	100 %
Die Qualität der Schnittstellen der zu integrierenden Anwendungs- systeme	Die Qualität einer verwendeten Schnittstelle ist hoch.	Die Qualität einer verwendeten Schnittstelle ist gering.	1	100 %	100 %	0 %	0 %
	Die Qualität der von einem zu integrierenden Anwendungssystem gelieferten Daten ist hoch.	Die Qualität der von einem zu integrierenden Anwendungssystem gelieferten Daten ist niedrig.	2	100 %	50 %	50 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Standardbasiertheit der Schnittstellen der zu integrierenden Anwendungssysteme	Eine Schnittstelle eines Anwendungssystems basiert auf Standards.	Eine Schnittstelle eines Anwendungssystems ist proprietär.	4	75 %	50 %	50 %	0 %
Die Änderungsdynamik der Schnittstellen der zu integrierenden Anwendungssysteme	Die verwendeten Schnittstellen sind konstant.	Die verwendeten Schnittstellen unterliegen einer hohen Änderungshäufigkeit.	1	100 %	100 %	0 %	0 %
Die Notwendigkeit der Anpassung der zu integrierenden Anwendungssysteme	Ein zu integrierendes Anwendungssystem muss nicht verändert werden.	Ein zu integrierendes Anwendungssystem muss angepasst werden.	2	50 %	100 %	0 %	0 %
Die Heterogenität der zu integrierenden Anwendungssysteme	Die zu integrierenden Anwendungssysteme basieren auf den gleichen Technologien.	Die zu integrierenden Anwendungssysteme basieren auf unterschiedlichen Technologien.	2	100 %	50 %	0 %	50 %
	Die zu integrierenden Anwendungssysteme sind hinsichtlich ihrer technischen und konzeptionellen Charakteristika einheitlich.	Die zu integrierenden Anwendungssysteme sind hinsichtlich ihrer technischen und konzeptionellen Charakteristika unterschiedlich.	1	100 %	0 %	100 %	0 %
	Die zu integrierenden Anwendungssysteme basieren auf einem einheitlichen Datenmodell.	Die zu integrierenden Anwendungssysteme basieren auf unterschiedlichen Datenmodellen.	3	100 %	33 %	67 %	0 %
	Es gibt ein homogenes Berechtigungskonzept über die zu integrierenden Anwendungssysteme hinweg.	Es gibt unterschiedliche Berechtigungskonzepte für die zu integrierenden Anwendungssysteme.	1	100 %	0 %	100 %	0 %
	Die Schnittstellen der zu integrierenden Anwendungssysteme sind gleichartig.	Die Schnittstellen der zu integrierenden Anwendungssysteme sind unterschiedlich.	1	100 %	100 %	0 %	0 %
Die Systemumgebung der zu integrierenden Anwendungssysteme	Ein zu integrierendes Anwendungssystem wird auf einer modernen Plattform betrieben.	Ein zu integrierendes Anwendungssystem wird auf einer unmodernen Plattform betrieben.	1	0 %	0 %	0 %	100 %
	Die zu integrierenden Anwendungssysteme sind unabhängig.	Die zu integrierenden Anwendungssysteme sind in eine komplexe gewachsene Anwendungslandschaft eingebunden.	1	0 %	0 %	0 %	100 %
	Die Anwendungsintegration betrifft lediglich die Verknüpfung von Servern im Rechenzentrum-Umfeld.	Die Anwendungsintegration erfolgt in einer komplexen Netzwerkumwelt. Die Anzahl der zu berücksichtigenden technischen Komponenten ist hoch.	1	100 %	0 %	100 %	0 %
	Die zu integrierenden Anwendungssysteme befinden sich in einer einheitlichen Systemumgebung.	Die zu integrierenden Anwendungssysteme befinden sich in unterschiedlichen Systemumgebungen.	1	100 %	100 %	0 %	0 %
	Das Umfeld der zu integrierenden Anwendungssysteme ist einheitlich oder muss bei der Integration nicht berücksichtigt werden.	Die zu integrierenden Anwendungssysteme besitzen heterogene Umfelder, die bei der Integration berücksichtigt werden müssen.	1	100 %	100 %	0 %	0 %

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rak- teristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Qualität der Dokumentation und der Ansprechpartner im Hinblick auf die zu integrierenden Anwendungssysteme	Die verfügbare Dokumentation ist quantitativ und qualitativ hinreichend.	Es ist keine technische oder fachliche Dokumentation verfügbar.	1	0 %	0 %	0 %	100 %
	Die vorgefundene Anwendungslandschaft ist definiert und dokumentiert.	Die vorgefundene Anwendungslandschaft ist nicht definiert und dokumentiert.	1	100 %	100 %	0 %	0 %
	Eine Schnittstelle eines Anwendungssystems ist sehr gut dokumentiert.	Eine Schnittstelle eines Anwendungssystems ist nicht dokumentiert.	4	100 %	75 %	25 %	0 %
	Es ist möglich, Personen zu finden, welche die zu integrierenden Anwendungssysteme genau kennen.	Es ist nicht möglich, Personen zu finden, welche die zu integrierenden Anwendungssysteme genau kennen.	1	100 %	0 %	100 %	0 %
	Es gibt klare vertragliche Regelungen mit dem Hersteller eines zu integrierenden Anwendungssystems.	Ein zu integrierendes Anwendungssystem ist eine alte Individualsoftware. Es gibt keinen klaren Ansprechpartner und kein Know-how für das Anwendungssystem.	1	100 %	100 %	0 %	0 %
	Es gibt eine gute Zusammenarbeit mit dem Hersteller eines zu integrierenden Anwendungssystems.	Der Hersteller eines zu integrierenden Anwendungssystems unterstützt das Integrationsvorhaben nicht.	2	100 %	100 %	0 %	0 %
Der Betrieb der zu integrierenden Anwendungssysteme	Die zu integrierenden Anwendungssysteme werden selbst betrieben.	Der Betrieb der zu integrierenden Anwendungssysteme ist an externe Dienstleister ausgelagert	1	0 %	0 %	100 %	0 %
	Die zu integrierenden Anwendungssysteme sind innerhalb der Organisation.	Die zu integrierenden Anwendungssysteme sind außerhalb der Organisation.	1	100 %	0 %	0 %	100 %
	Es herrscht Informationstransparenz im Hinblick auf Betrieb und Nutzung eines zu integrierenden Anwendungssystems.	Es besteht ein Informationsdefizit im Hinblick auf Betrieb und Nutzung eines zu integrierenden Anwendungssystems.	1	100 %	0 %	0 %	100 %

Konstrukte zur Gestaltung der Integrationslösung

Dem Themenbereich der Gestaltung der Integrationslösung wurden 19 Gruppen zugeordnet, die insgesamt 49 Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rakte- ristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Investition in Entwurf und Archi- tektur der Inte- grationslösung	Die Gruppe der Projektmitarbeiter, die mit der Entwicklung der Architektur betraut sind, ist groß und spielt eine wichtige Rolle im Projekt.	Die Gruppe der Projektmitarbeiter, die mit der Entwicklung der Archi- tektur betraut sind, ist klein und hat lediglich eine beratende Funktion.	1	0 %	100 %	0 %	0 %
	Bei der Entwicklung wird eine architektonische Durchgängigkeit sichergestellt.	Es werden lediglich rudimentäre architektonische Vorgaben ge- macht.	1	0 %	100 %	0 %	0 %
	Fachliches Design, technisches Design und Schnittstellendesign der Integrationslösung werden zentral erstellt und konsequent umgesetzt.	Fachliches Design, technisches Design und Schnittstellendesign der Integrationslösung werden nicht zentral erstellt oder nicht eingehalten.	1	0 %	100 %	0 %	0 %
Die Komplexität der Integrationsarchi- tektur	Die Komplexität der Integrations- architektur ist niedrig.	Die Komplexität der Integrations- architektur ist groß.	2	100 %	50 %	50 %	0 %
	Die Integrationsarchitektur be- steht aus wenigen Komponenten.	Die Integrationsarchitektur besteht aus vielen Komponenten.	2	100 %	0 %	50 %	50 %
	Die Integrationsarchitektur be- steht aus einer Schicht ("one tier architecture").	Die Integrationsarchitektur besteht aus mehreren Schichten ("multi tier architecture").	1	100 %	0 %	100 %	0 %
Die Änderungsdyn- amik der Inte- grationsarchitektur	Die Integrationsarchitektur ist klar geplant.	Die Integrationsarchitektur ist im Fluss.	1	100 %	100 %	0 %	0 %
	Es gibt keine Änderung der Integrationsarchitektur während der Projektlaufzeit.	Die Integrationsarchitektur ändert sich während der Projektlaufzeit.	1	100 %	100 %	0 %	0 %
Die Struktur der Kommunikations- beziehungen zwi- schen den zu inte- grierenden An- wendungssystemen	Zwischen den zu integrierenden Anwendungssystemen besteht eine 1-zu-1-Beziehung.	Zwischen den zu integrierenden Anwendungssystemen besteht eine n-zu-m-Beziehung.	1	100 %	100 %	0 %	0 %
	Die Kommunikation zwischen den zu integrierenden Anwen- dungssystemen ist unidirektional.	Die Kommunikation zwischen den zu integrierenden Anwendungssys- temen ist bidirektional.	1	0 %	0 %	100 %	0 %
	Im Rahmen der Integrationslö- sung spielen mehrere Anwen- dungssysteme eine zentrale Rolle.	Im Rahmen der Integrationslösung spielt ein Anwendungssystem eine zentrale Rolle. Die Anbindung anderer Anwendungssysteme orientiert sich an diesem zentralen Anwendungssystem.	1	100 %	0 %	0 %	100 %
	Für die Integration der Anwen- dungssysteme kann ein zentraler Hub eingesetzt werden.	Für die Integration der Anwen- dungssysteme kann kein zentraler Hub eingesetzt werden.	1	100 %	100 %	0 %	0 %
Die Anzahl und die Komplexität der Schnittstellen	Die Anzahl der Schnittstellen ist gering.	Die Anzahl der Schnittstellen ist groß.	6	100 %	67 %	33 %	0 %
	Die Komplexität einer Schnittstel- le ist gering.	Die Komplexität einer Schnittstelle ist groß.	4	75 %	25 %	50 %	25 %
	Die Anzahl an Datensätzen, die eine Schnittstelle passieren, ist gering.	Die Anzahl an Datensätzen, die eine Schnittstelle passieren, ist groß.	1	100 %	100 %	0 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Komplexität der Adapter	Die Komplexität eines Adapters ist gering.	Die Komplexität eines Adapters ist groß.	1	100 %	0 %	100 %	0 %
Die Anzahl der Formate der Kommunikation unter den zu integrierenden Anwendungssystemen	Im Rahmen der Interaktion von zu integrierenden Anwendungssystemen sind wenige Systemgrenzen zu überwinden und Verarbeitungsschritte durchzuführen.	Im Rahmen der Interaktion von zu integrierenden Anwendungssystemen sind viele Protokoll- und Formatsprünge zu überwinden.	1	0 %	100 %	0 %	0 %
	Die Anzahl der Datensatzformate ist gering.	Die Anzahl der Datensatzformate ist groß.	1	100 %	0 %	100 %	0 %
	Die Anzahl der Nachrichtenformate ist gering.	Die Anzahl der Nachrichtenformate ist groß.	1	100 %	100 %	0 %	0 %
	Die Integration aller Anwendungssysteme basiert auf den gleichen Kommunikationsarten und -formaten.	Die Integration der Anwendungssysteme basiert auf vielen unterschiedlichen Kommunikationsarten und -formaten.	1	100 %	100 %	0 %	0 %
Die Anzahl und die Komplexität der Transformationen	Die Anzahl der zu realisierenden Transformationen ("Mappings") ist gering.	Die Anzahl der zu realisierenden Transformationen ("Mappings") ist groß.	3	100 %	67 %	33 %	0 %
	Die Komplexität einer zu realisierenden Transformation ("Mapping") ist gering.	Die Komplexität einer zu realisierenden Transformation ("Mapping") ist groß.	1	100 %	100 %	0 %	0 %
Die Komplexität der Datenstrukturen der Integrationslösung	Die Komplexität des Datenmodells ist gering.	Die Komplexität des Datenmodells ist groß.	1	100 %	0 %	100 %	0 %
	Die Anzahl der Datenbanktabellen ist gering.	Die Anzahl der Datenbanktabellen ist groß.	1	0 %	0 %	0 %	100 %
	Die Komplexität einer Datenbanktabelle ist gering.	Die Komplexität einer Datenbanktabelle ist groß.	1	100 %	0 %	0 %	100 %
	Die Anzahl der Datenelemente ist gering.	Die Anzahl der Datenelemente ist groß.	1	100 %	0 %	0 %	100 %
	Die Komplexität eines Datenelements ist gering.	Die Komplexität eines Datenelements ist groß.	1	0 %	0 %	0 %	100 %
	Die Informationsstrukturen sind flach.	Die Informationsstrukturen sind baumartig verschachtelt.	1	100 %	0 %	100 %	0 %
	Die Komplexität der Struktur eines Datenobjekts ist gering.	Die Komplexität der Struktur eines Datenobjekts ist groß.	1	0 %	100 %	0 %	0 %
Der Einsatz eines kanonischen Datenmodell für die Durchführung von Transformationen	Die Transformationen ("Mappings") werden anhand eines zentralen ("kanonischen") Datenmodells realisiert.	Die Transformationen ("Mappings") werden individuell zwischen den Anwendungssystemen realisiert.	1	100 %	0 %	100 %	0 %
	Es kann ein einziges zentrales ("kanonisches") Datenmodell für die zu integrierenden Anwendungssysteme gefunden werden.	Mehrere zentrale ("kanonische") Datenmodelle, die jeweils für Teilmengen der zu integrierenden Anwendungssysteme gelten, müssen miteinander vereinbart werden.	1	100 %	0 %	100 %	0 %
	Es muss kein zentrales ("kanonisches") Datenmodell geschaffen werden.	Es muss ein zentrales ("kanonisches") Datenmodell geschaffen werden.	1	100 %	100 %	0 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die mit der Integrationslösung zu überwindenden Netzwerkgrenzen	Die Integration der Anwendungssysteme erfolgt über wenige Netzwerkgrenzen hinweg.	Die Integration der Anwendungssysteme erfolgt über viele Netzwerkgrenzen hinweg.	1	100 %	0 %	100 %	0 %
	Die Integration der Anwendungssysteme erfolgt innerhalb einer demilitarisierten Zone/Sicherheitszone.	Die Integration der Anwendungssysteme erfolgt über mehrere demilitarisierten Zonen/Sicherheitszonen hinweg.	1	100 %	0 %	100 %	0 %
Die Gestaltung der Kommunikation zwischen den Anwendungssystemen	Die Integration der Anwendungssysteme erfolgt auf der Basis synchroner Mechanismen ("request-reply").	Die Integration der Anwendungssysteme erfolgt auf der Basis asynchroner Mechanismen ("public-subscribe").	1	100 %	0 %	0 %	100 %
	Die Kommunikation der zu integrierenden Anwendungssysteme basiert auf Bulk-Lieferungen.	Die Kommunikation der zu integrierenden Anwendungssysteme basiert auf Einzelsatzverarbeitung.	1	100 %	0 %	100 %	0 %
Das Ausmaß der Kopplung zwischen den Komponenten der Integrationslösung	Die zu integrierenden Anwendungssysteme werden lose gekoppelt.	Die zu integrierenden Anwendungssysteme werden eng gekoppelt.	2	100 %	50 %	50 %	0 %
	Bei der Gestaltung der Integrationslösung wird keine lose Kopplung zwischen den zu integrierenden Anwendungssystemen und den eingesetzten Integrationskomponenten angestrebt. Die Anwendungssysteme "kennen" die verwendete Technologie.	Bei der Gestaltung der Integrationslösung wird eine lose Kopplung zwischen den zu integrierenden Anwendungssystemen und den eingesetzten Integrationskomponenten angestrebt ("Transparenz der Technologie").	1	100 %	0 %	0 %	100 %
	Die Integration der Anwendungssysteme erfolgt durch die Nutzung direkter Schnittstellen.	Die Integration der Anwendungssysteme erfolgt durch den Einsatz einer Middleware.	1	100 %	0 %	0 %	100 %
Die Realisierung der Steuerung der Interaktionen der zu integrierenden Anwendungssysteme	Geschäftsprozesse werden nicht auf der Integrationsplattform abgebildet. Die Steuerung der Interaktionen zwischen den zu integrierenden Anwendungssystemen erfolgt durch die Anwendungssysteme selbst.	Geschäftsprozesse werden auf der Integrationsplattform abgebildet. Die Steuerung der Interaktionen zwischen den zu integrierenden Anwendungssystemen erfolgt über die Integrationsplattform.	1	100 %	0 %	100 %	0 %
	Die in der Integrationslösung abzubildenden Geschäftsprozesse und Geschäftsregeln werden fest im Programmcode umgesetzt.	Die in der Integrationslösung abzubildenden Geschäftsprozesse und Geschäftsregeln werden extrahiert und beispielsweise als "business rules" implementiert.	2	100 %	0 %	50 %	50 %
Die Anzahl und die Komplexität der zu entwickelnden Module der Integrationslösung	Die Anzahl der zu entwickelnden Module ist gering.	Die Anzahl der zu entwickelnden Module ist groß.	1	0 %	0 %	0 %	100 %
	Die Komplexität eines zu entwickelnden Moduls ist gering.	Die Komplexität eines zu entwickelnden Moduls ist groß.	1	0 %	0 %	0 %	100 %
Die Anzahl und die Komplexität der zu realisierenden Funktionen	Die Anzahl der zu realisierenden Funktionen ist gering.	Die Anzahl der zu realisierenden Funktionen ist groß.	1	100 %	0 %	0 %	100 %
	Die Komplexität einer zu realisierenden Funktion ist gering.	Die Komplexität einer zu realisierenden Funktion ist groß.	1	0 %	0 %	0 %	100 %
Die Gestaltung der Authentifizierung der Benutzer	Eine Authentifizierung erfolgt mittels eines geheimen Schlüssels, d. h. ohne eine Passwortsynchronisation.	Eine Authentifizierung erfolgt über Benutzer- und Passwortsynchronisation.	1	100 %	0 %	100 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Berücksichtigung der Fachlichkeit bei der Gestaltung der Integrationslösung	Die Integrationslösung ist an die Geschäftsprozesse und die Betriebsführung der IT des Kunden angepasst.	Die Integrationslösung ist nicht an die Geschäftsprozesse und die Betriebsführung der IT des Kunden angepasst.	2	0 %	100 %	0 %	0 %
	Das Zusammenspiel von Technologie und fachlichen Prozessen ist verstanden und kommuniziert.	Das Zusammenspiel von Technologie und fachlichen Prozessen ist nicht verstanden und kommuniziert.	1	100 %	0 %	100 %	0 %
Die Berücksichtigung der Gleichzeitigkeit von Geschäftsprozessen bei der Gestaltung der Integrationslösung	Im Falle der Integration auf Datenebene und der Gleichzeitigkeit von Geschäftsprozessen wird der Status vor der Persistierung von Daten geprüft und plausibilisiert.	Im Falle der Integration auf Datenebene und der Gleichzeitigkeit von Geschäftsprozessen wird der Status vor der Persistierung von Daten nicht geprüft und plausibilisiert.	1	100 %	0 %	100 %	0 %

Konstrukte zu den eingesetzten Integrationsprodukten, Technologien und Entwicklungswerkzeugen

Dem Themenbereich der eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge wurden 10 Gruppen zugeordnet, die insgesamt 29 Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Verfügbarkeit von Integrationsprodukten	Die Integrationslösung basiert auf standardisierten Integrationsprodukten.	Die Integrationslösung wird vollständig eigenentwickelt.	4	100 %	50 %	25 %	25 %
	Die Integrationslösung basiert auf konfigurierbaren Integrationsprodukten.	Die Integrationslösung wird mit Entwicklungswerkzeugen umgesetzt, die keine fertige Lösung produzieren.	1	0 %	0 %	100 %	0 %
	Moderne Integrationsprodukte sind verfügbar und geben die Integrationslösung vor.	Es mangelt an Unterstützung der Integrationsaufgaben durch Integrationsprodukte.	1	100 %	100 %	0 %	0 %
	Ein Standardadapter steht zur Verfügung.	Ein Adapter muss selber entwickelt werden.	1	100 %	100 %	0 %	0 %
Die Schwierigkeit des Verstehens und Verwendens der eingesetzten Integrationsprodukte	Das eingesetzte Integrationsprodukt basiert auf generischen Standards.	Das eingesetzte Integrationsprodukt ist proprietär und sehr speziell.	1	100 %	0 %	100 %	0 %
	Das eingesetzte Integrationsprodukt kann sehr leicht installiert/aufgesetzt werden.	Das eingesetzte Integrationsprodukt ist schwer zu installieren/aufzusetzen.	1	0 %	0 %	0 %	100 %
Die Qualität der eingesetzten Integrationsprodukte	Der Reifegrad des eingesetzten Integrationsproduktes ist hoch.	Der Reifegrad des eingesetzten Integrationsproduktes ist niedrig.	2	50 %	50 %	50 %	0 %
	Das eingesetzte Integrationsprodukt ist etabliert und gut dokumentiert.	Das eingesetzte Integrationsprodukt ist nicht etabliert und wenig dokumentiert.	1	0 %	100 %	0 %	0 %
	Das eingesetzte Integrationsprodukt ist stabil und zuverlässig.	Das eingesetzte Integrationsprodukt ist instabil und unberechenbar.	1	0 %	100 %	0 %	0 %
	Das eingesetzte Integrationsprodukt kann standardmäßig in der Integrationslösung eingesetzt werden.	Die Standardschnittstellen des eingesetzten Integrationsproduktes sind fehlerhaft.	1	100 %	100 %	0 %	0 %
Die Änderungsdynamik der eingesetzten Integrationsprodukte	Es sind keine Updates des eingesetzten Integrationsproduktes erforderlich.	Das eingesetzte Integrationsprodukt weist kurze Updateintervalle auf.	1	100 %	0 %	100 %	0 %
	Es ist kein Versionswechsel des eingesetzten Integrationsproduktes während der Projektlaufzeit erforderlich.	Es ist ein Versionswechsel des eingesetzten Integrationsproduktes während der Projektlaufzeit erforderlich.	1	100 %	0 %	100 %	0 %

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rak- teristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Eignung der eingesetzten Integ- rationsprodukte für die Integrationslö- sung	Es werden geeignete Integrati- onsprodukte in der Integrationslö- sung eingesetzt.	Es werden ungeeignete Integrati- onsprodukte in der Integrationslö- sung eingesetzt.	1	100 %	100 %	0 %	0 %
	Das eingesetzte Integrationspro- dukt erfüllt die gestellten funktio- ns- und qualitätsbezogenen Anforderungen.	Das eingesetzte Integrationsprodukt erfüllt die gestellten funktions- und qualitätsbezogenen Anforderungen nicht.	2	50 %	0 %	100 %	0 %
	Es ist keine funktionale Erweite- rung des eingesetzten Integrati- onsproduktes notwendig.	Es werden funktionale Erweiterun- gen des eingesetzten Integrations- produktes durch dessen Hersteller erforderlich.	1	100 %	0 %	100 %	0 %
	Das eingesetzte Integrationspro- dukt bietet tatsächlich die von dem Hersteller genannten Fea- tures.	Das eingesetzte Integrationsprodukt bietet nicht die vom Hersteller genannten Features.	1	100 %	0 %	100 %	0 %
	Das eingesetzte Integrationspro- dukt unterstützt standardmäßig ein erforderliches Nachrichten- format oder eine anzubindende Schnittstelle.	Das eingesetzte Integrationsprodukt unterstützt ein erforderliches Nachrichtenformat oder eine anzubindende Schnittstelle nicht standardmäßig.	1	100 %	0 %	100 %	0 %
Die Abhängigkeit von den Herstellern der eingesetzten Integrations- produkte	Es gibt ein starkes Lieferanten- management bezüglich der einge- setzten Integrationsprodukte.	Es gibt ein schwaches Lieferan- tenmanagement bezüglich der eingesetzten Integrationsprodukte. Man gerät gegenüber Lieferanten in Abhängigkeiten.	1	100 %	0 %	100 %	0 %
	Das eingesetzte Integrationspro- dukt stammt aus einer Hand.	Bei dem eingesetzten Integrations- produkt bestehen Abhängigkeiten von Drittherstellern.	1	100 %	0 %	0 %	100 %
Die Unterstützung durch die Hersteller der eingesetzten Integrations- produkte	Der Hersteller des eingesetzten Integrationsproduktes ist etabliert und hat entsprechende Support- möglichkeiten.	Der Hersteller des eingesetzten Integrationsproduktes ist klein und hat nur geringe Supportmöglichkei- ten.	1	100 %	0 %	0 %	100 %
Die eingesetzten Technologien	Die Anzahl der eingesetzten Technologien ist niedrig.	Die Anzahl der eingesetzten Tech- nologien ist groß.	1	0 %	0 %	100 %	0 %
	Die eingesetzte Technologie lässt sich durch die Projektmitarbeiter leicht anwenden.	Die eingesetzte Technologie lässt sich durch die Projektmitarbeiter schwer anwenden.	1	100 %	100 %	0 %	0 %
	Die eingesetzte Technologie ist etabliert.	Die eingesetzte Technologie ist neuartig.	3	0 %	33 %	33 %	33 %
	Die eingesetzte Technologie bietet Unterstützung durch Stan- dardbibliotheken.	Man muss vollständig eigenentwik- keln.	1	0 %	0 %	100 %	0 %
	Die Integrationslösung basiert auf programmatischer Technologie (d. h. Realisierung mittels Pro- grammiersprache).	Die Integrationslösung basiert auf deklarativer Technologie (d. h. Realisierung mittels Auszeich- nungssprachen, wie XML).	1	100 %	0 %	0 %	100 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Charakteristisch	Anteil der Beurteilungen		
	Weniger Aufwand wenn...	Höherer Aufwand wenn...			Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Unterstützung durch Entwicklungswerkzeuge	Es wird ein vorhandenes Entwicklungswerkzeug eingesetzt.	Es wird ein integriertes Entwicklungswerkzeug für die Integrationslösung entwickelt.	1	0 %	0 %	100 %	0 %
	Moderne Entwicklungswerkzeuge sind verfügbar.	Moderne Entwicklungswerkzeuge sind nicht verfügbar.	2	0 %	0 %	50 %	50 %
	Die Services einer serviceorientierten Architektur werden aufgrund eines generischen Ansatzes durch ein Entwicklungswerkzeug generiert.	Die Services einer serviceorientierten Architektur werden jeweils eigenentwickelt.	1	100 %	100 %	0 %	0 %
Die Einhaltung der Standards der Anwendungssysteme	Die Integration eines ERP-Systems erfolgt unter Einhaltung dessen Standards.	Die Integration eines ERP-Systems erfolgt durch Eigenentwicklung bei Abweichung vom Standard des ERP-Systems.	1	0 %	0 %	100 %	0 %

Konstrukte zum Testen der Integrationslösung und die Überleitung in die Nutzung

Dem Themenbereich des Testens der Integrationslösung und der Überleitung in die Nutzung wurden 5 Gruppen zugeordnet, die insgesamt 17 der persönlichen Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Der Umfang des Testens	Es werden wenige Teststufen durchgeführt.	Es werden viele Teststufen durchgeführt.	1	100 %	0 %	100 %	0 %
	Es werden wenige Regressionstests durchgeführt.	Es werden viele Regressionstests durchgeführt.	1	0 %	0 %	0 %	100 %
	Es werden wenige Testfälle durchgeführt.	Es werden viele Testfälle durchgeführt.	1	0 %	0 %	100 %	0 %
Die Sorgfalt des Testens	Es gibt eine hohe Testfallüberdeckung bezüglich der Schnittstellen. Es gibt eine lange Testphase.	Es gibt eine geringe Testfallüberdeckung bezüglich der Schnittstellen. Es gibt eine kurze Testphase.	1	0 %	100 %	0 %	0 %
	Es wird sehr umfangreich und systematisch getestet.	Es wird nicht sehr umfangreich und systematisch getestet.	1	0 %	100 %	0 %	0 %
	Es gibt eine klare zeitliche Einplanung der Integrationstestphase.	Es gibt keine klare zeitliche Einplanung der Integrationstestphase.	1	0 %	0 %	100 %	0 %
	Es gibt ein definiertes Testvorgehen.	Es gibt kein definiertes Testvorgehen.	1	0 %	0 %	100 %	0 %
Die herrschenden Umstände zur Durchführung des Testens	Eine komplexe Testumgebung muss nicht aufgebaut werden.	Eine komplexe Testumgebung muss aufgebaut werden.	1	0 %	0 %	100 %	0 %
	Testsysteme der zu integrierenden Anwendungssysteme sind vorhanden.	Die zu integrierenden Anwendungssysteme müssen für Entwicklung und Test simuliert werden.	1	100 %	0 %	0 %	100 %
	Testdaten für die zu integrierenden Anwendungssysteme sind vorhanden.	Testdaten für die zu integrierenden Anwendungssysteme sind nicht vorhanden.	1	100 %	0 %	100 %	0 %
	Die Tests sind vollständig automatisiert.	Die Tests sind vollständig manuell.	1	0 %	0 %	0 %	100 %
	Erforderliche Korrekturen können unmittelbar vorgenommen werden. Entwicklung und Test durchdringen einander.	Es gibt lange Verzögerungszeiten ("turn-around-time") um Korrekturen vornehmen zu können.	1	0 %	0 %	100 %	0 %
	Die eingesetzte Technologie ist Interpreter-basiert. Man kann im laufenden Betrieb Änderungen vornehmen und sofort testen.	Die eingesetzte Technologie ist compiler-basiert. Man muss nach einer Änderung Testdurchläufe starten.	1	0 %	100 %	0 %	0 %
Die Verteilung der Zuständigkeiten beim Testen	Die Zuständigkeit für den Test der gesamten Integrationslösung ist gebündelt.	Es gibt getrennte Zuständigkeiten für den Test von Teilen der Integrationslösung.	2	50 %	50 %	0 %	50 %
	Es gibt ein internes und vollständig verstandenes Testsystem.	Es gibt verteilte Testsysteme, die getrennt betreut werden.	1	100 %	100 %	0 %	0 %
Das Vorgehen bei der Überleitung in die Nutzung	Die Anzahl der durchzuführenden Trainings ist klein.	Die Anzahl der durchzuführenden Trainings ist groß.	1	0 %	0 %	0 %	100 %
	Der Auslieferungsprozess ist einfach.	Der Auslieferungsprozess ist komplex.	1	0 %	0 %	100 %	0 %

Konstrukte zum Kunden und den Stakeholdern der Integrationslösung

Dem Themenbereich des Kunden und der Stakeholder des Integrationsvorhabens wurden 7 Gruppen zugeordnet, die insgesamt 34 Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rak- teristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Beziehung zum Kunden	Man führt das Integrationsvorhaben als externer Dienstleister für einen Kunden durch.	Man führt das Integrationsvorhaben als interner Dienstleister durch.	1	100 %	0 %	0 %	100 %
	Es gibt einen klaren Arbeitsauftrag und eine klare Trennung von Auftraggeber und Auftragnehmer.	Es gibt keinen klaren Arbeitsauftrag und der Kunde ist in die Entwicklung der Integrationslösung involviert.	1	0 %	100 %	0 %	0 %
	Man steht seit langem in Beziehung zum Kunden des Integrationsvorhabens.	Das Integrationsvorhaben stellt eine erstmalige oder einmalige Beziehung zum Kunden dar.	1	0 %	0 %	0 %	100 %
	Die Integrationslösung wird lediglich entwickelt und bereitgestellt.	Die Projektmitarbeiter leisten Unterstützung ("Support") für die Betriebsseite.	1	0 %	0 %	0 %	100 %
	Der Kunde hat Erfahrung darin, Entwicklungsvorhaben gemeinsam mit externen Dienstleistern durchzuführen.	Der Kunde kauft IT-Lösungen als Black-Box ("wie einen PC").	1	0 %	0 %	0 %	100 %
Die Homogenität der Stakeholder des Integrationsvorhabens	Die Anzahl der beteiligten Parteien ist gering.	Die Anzahl der beteiligten Parteien ist groß.	3	0 %	33 %	67 %	0 %
	Das Integrationsvorhaben betrifft Anwendungssysteme nur einer organisatorischen Einheit.	Das Integrationsvorhaben betrifft Anwendungssysteme mehrerer organisatorischer Einheiten.	1	100 %	0 %	100 %	0 %
	Das Integrationsvorhaben betrifft Anwendungssysteme eines Unternehmens.	Das Integrationsvorhaben betrifft Anwendungssysteme mehrerer Unternehmen.	1	100 %	0 %	100 %	0 %
	Das Integrationsvorhaben betrifft nur Anwendungssysteme des Kunden.	Das Integrationsvorhaben betrifft Anwendungssysteme des Kunden sowie Anwendungssysteme externer Partner des Kunden.	1	100 %	100 %	0 %	0 %
Die Einstellung der Stakeholder zum Integrationsvorhaben	Die Stakeholder des Integrationsvorhabens haben eine gemeinsame Vision.	Die Stakeholder des Integrationsvorhabens haben unterschiedliche Zielsetzungen.	1	0 %	100 %	0 %	0 %
	Das Integrationsvorhaben wird durch die Benutzer der zu integrierenden Systeme unterstützt.	Es gibt eine Abwehrhaltung der Benutzer der zu integrierenden Systeme gegen das Integrationsvorhaben.	1	0 %	100 %	0 %	0 %
	Die Stakeholder des Integrationsvorhabens haben keine Angst vor damit verbundenen Veränderungen.	Die Stakeholder des Integrationsvorhabens haben Angst vor damit verbundenen Veränderungen.	1	100 %	0 %	100 %	0 %
	Die zu entwickelnde Integrationslösung ist vollständig neuartig.	Die zu entwickelnde Integrationslösung konkurriert gegen bestehende Lösungen beim Kunden.	1	100 %	0 %	100 %	0 %

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rakte- ristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Ausprägung der Kommunikation mit dem Kunden	Die Anforderungen können unmittelbar mit dem Endkunden abgestimmt werden.	Die Anforderungen des Endkunden werden über Dritte kommuniziert.	1	0 %	0 %	0 %	100 %
	Es besteht ausschließlich ein formalisierter Kontakt zwischen den Entwicklern und den Fachabteilungen.	Die Fachabteilungen können die Entwickler direkt kontaktieren.	1	0 %	0 %	0 %	100 %
	Es gibt eine starke und formale Einbeziehung der Fachabteilungen.	Die Einbeziehung der Fachabteilungen erfolgt spontan.	1	0 %	0 %	100 %	0 %
	Es erfolgt eine fortwährende Kommunikation von Zwischenergebnissen gegenüber dem Kunden.	Dem Kunden werden keine Zwischenergebnisse kommuniziert.	1	0 %	0 %	100 %	0 %
Das Ausmaß der Unterstützung durch den Kunden	Es gibt eine starke Kundenbeteiligung und -mitarbeit.	Es gibt eine sehr geringe Kundenbeteiligung und -mitarbeit.	1	0 %	0 %	100 %	0 %
	Der Kunde liefert fachliche Unterstützung bei der Realisierung.	Der Kunde ist nicht in die Realisierung eingebunden.	1	0 %	0 %	100 %	0 %
	Der Kunde kann seine Mitwirkungspflicht erfüllen.	Der Kunde kann seine Mitwirkungspflicht nicht erfüllen.	1	0 %	0 %	0 %	100 %
	Die Projektmitarbeiter werden durch die Endanwender unterstützt.	Die Projektmitarbeiter werden nicht durch die Endanwender unterstützt.	2	50 %	50 %	50 %	0 %
	Die Management Attention für das Integrationsvorhaben ist hoch.	Die Management Attention für das Integrationsvorhaben ist niedrig.	2	0 %	0 %	100 %	0 %
Das Ausmaß von Erfahrung und Wissen des Kunden	Der Kunde hat Erfahrung mit der Anwendungsintegration und verfügt über ein klares und einheitliches Begriffssystem zu deren Beschreibung.	Der Kunde hat keine Erfahrung mit der Anwendungsintegration und verfügt über kein klares und einheitliches Begriffssystem zu deren Beschreibung.	1	0 %	0 %	100 %	0 %
	Der Kunde hat sich mit der Anwendungsintegration beschäftigt und kennt sich damit aus.	Der Kunde muss bezüglich der Anwendungsintegration "missioniert" werden.	1	100 %	0 %	0 %	100 %
	Der Erkenntnisgrad des Managements bezüglich der Probleme von Integrationsvorhaben ist hoch.	Der Erkenntnisgrad des Managements bezüglich der Probleme von Integrationsvorhaben ist gering.	1	100 %	0 %	0 %	100 %
	Die Business-Stakeholder besitzen ein hohes IT-Verständnis.	Die Business-Stakeholder besitzen kein IT-Verständnis.	1	100 %	0 %	0 %	100 %
	Der Kunde ist mit den zu integrierenden Anwendungssystemen vertraut.	Der Kunde ist mit den zu integrierenden Anwendungssystemen nicht vertraut.	1	0 %	100 %	0 %	0 %
	Der Kunde ist fachlich und technisch sehr kompetent.	Der Kunde ist fachlich und technisch eingeschränkt kompetent.	1	0 %	0 %	100 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Gestaltung der Zuständigkeiten auf Kundenseite	Es gibt einen verantwortlichen Ansprechpartner auf Kundenseite.	Es gibt viele verschiedene Ansprechpartner auf Kundenseite.	4	50 %	50 %	25 %	25 %
	Man hat unmittelbaren Kontakt zu den Entscheidungsträgern des Kunden.	Die Ansprechpartner beim Kunden haben keine Entscheidungsbefugnis.	1	0 %	0 %	0 %	100 %
	Im Laufe des Integrationsvorhabens ist keine weitere Entscheidung des Kunden notwendig.	Der Kunde hat keine klaren und kompakten Entscheidungsstrukturen. Keiner der Ansprechpartner traut sich eine Entscheidung zu treffen.	1	0 %	100 %	0 %	0 %
	Es existiert eine etablierte, mehrheitlich demokratische Entscheidungsstruktur beim Kunden.	Es existiert eine autokratische Entscheidungsstruktur beim Kunden.	1	0 %	0 %	100 %	0 %
	Beim Kunden gibt es eine zentrale Leitung der IT.	Beim Kunden erfolgt die Leitung der IT dezentral.	1	100 %	100 %	0 %	0 %
	Beim Kunden gibt es eine starke Leitung der IT.	Beim Kunden gibt es eine schwache Leitung der IT.	1	0 %	0 %	100 %	0 %

Konstrukte zu den Projektmitarbeitern

Dem Themenbereich der Projektmitarbeiter wurden 9 Gruppen zugeordnet, die insgesamt 32 Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rak- teristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Anzahl der Projektmitarbeiter	Es gibt wenige Projektmitarbeiter.	Es gibt viele Projektmitarbeiter.	4	0 %	50 %	50 %	0 %
Die allgemeine Produktivität der Projektmitarbeiter	Die Produktivität der Projektmitarbeiter ist hoch.	Die Produktivität der Projektmitarbeiter ist niedrig.	1	0 %	0 %	100 %	0 %
Die Homogenität der Projektmitarbeiter	Die Projektmitarbeiter bilden ein homogenes Team mit einem gemeinsamen Verständnis von Begriffen und Konzepten.	Die Projektmitarbeiter bilden lose Teamverbände ohne ein gemeinsames Verständnis von Begriffen und Konzepten.	1	0 %	100 %	0 %	0 %
	Die Projektmitarbeiter entstammen demselben Unternehmen und bilden ein homogenes Team.	Die Projektmitarbeiter entstammen unterschiedlichen Unternehmen und bilden ein heterogenes Team.	1	100 %	0 %	100 %	0 %
	Die Projektmitarbeiter formen ein kleines Spezialistenteam.	Die Projektmitarbeiter formen sehr heterogene Teamstrukturen.	1	100 %	0 %	0 %	100 %
Erfahrung und Wissen der Projektmitarbeiter	Es ist kein tiefgehendes technisches Spezialwissen erforderlich.	Punktuell ist ein tiefgehendes technisches Spezialwissen erforderlich.	1	100 %	100 %	0 %	0 %
	Kompetente Projektmitarbeiter sind verfügbar.	Kompetente Projektmitarbeiter sind nicht verfügbar.	1	0 %	0 %	100 %	0 %
	Personen mit Spezialisten-Know-how sind verfügbar.	Personen mit Spezialisten-Know-how sind nicht verfügbar.	1	0 %	0 %	0 %	100 %
	Die Projektmitarbeiter haben sehr gutes Fachwissen.	Die Projektmitarbeiter haben kein Fachwissen.	4	50 %	50 %	50 %	0 %
	Die Projektmitarbeiter sind mit den zu integrierenden Anwendungssystemen vertraut.	Die Projektmitarbeiter sind nicht mit den zu integrierenden Anwendungssystemen vertraut.	2	100 %	0 %	50 %	50 %
	Die Projektmitarbeiter sind erfahren.	Die Projektmitarbeiter sind unerfahren.	4	25 %	100 %	0 %	0 %
	Die mit der Analyse betrauten Projektmitarbeiter haben sehr gute Software Engineering Skills ("hard skills").	Die mit der Analyse betrauten Projektmitarbeiter haben keine Software Engineering Skills ("hard skills").	1	0 %	100 %	0 %	0 %
	Die Projektmitarbeiter sind mit der eingesetzten Technologie vertraut.	Die Projektmitarbeiter sind mit der eingesetzten Technologie nicht vertraut.	1	100 %	100 %	0 %	0 %
	Die Projektmitarbeiter sind Experten im Umgang mit dem eingesetzten Integrationsprodukt.	Die Projektmitarbeiter wissen nicht, welche Features des eingesetzten Integrationsproduktes in der Problemdomäne eingesetzt werden können.	2	50 %	0 %	100 %	0 %
	Die Anwendungssysteme und Geschäftsprozesse sind den mit dem Testen betrauten Projektmitarbeitern bekannt.	Die Anwendungssysteme und Geschäftsprozesse sind den mit dem Testen betrauten Projektmitarbeitern nicht bekannt.	1	0 %	0 %	100 %	0 %
	Die Projektbeteiligten besitzen von Beginn an das erforderliche Wissen.	Die Projektbeteiligten müssen das erforderliche Wissen während der Projektlaufzeit aufbauen.	1	0 %	0 %	100 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Qualität der Zusammenarbeit unter den Projektmitarbeitern	Die Projektmitarbeiter haben eine "Open Source Philosophie" gegenüber anderen Projektmitarbeitern.	Die Projektmitarbeiter gewähren anderen Projektmitarbeitern keinen Einblick in den von ihnen entwickelten Programmcode.	1	0 %	0 %	100 %	0 %
	Unter den Projektbeteiligten herrscht Teamgeist.	Unter den Projektbeteiligten herrscht kein Teamgeist.	1	0 %	0 %	100 %	0 %
	Die mit der Analyse betrauten Projektmitarbeiter besitzen sehr hohe soziale Kompetenz ("soft skills").	Die mit der Analyse betrauten Projektmitarbeiter sind nicht teamfähig.	1	0 %	100 %	0 %	0 %
Die Kontinuität der Projektmitarbeiter	Das Team der Projektmitarbeiter besteht seit längerem.	Das Team der Projektmitarbeiter muss wiederholt aus verschiedenen Quellen zusammengestellt werden.	1	0 %	0 %	100 %	0 %
	Die Projektmitarbeiter formen ein eingespieltes Team.	Die Projektmitarbeiter sind nicht miteinander vertraut.	1	0 %	100 %	0 %	0 %
	Es gibt dedizierte Mitarbeiter für das Integrationsvorhaben.	Die Mitarbeiter arbeiten parallel auch an projektexternen Aufgaben. Die Ansprechpartner für das Integrationsvorhaben wechseln.	1	0 %	100 %	0 %	0 %
Die Einstellung der Projektmitarbeiter zum Integrationsvorhaben	Die Projektmitarbeiter sind motiviert.	Die Projektmitarbeiter sind nicht motiviert.	1	0 %	100 %	0 %	0 %
	Die Projektmitarbeiter lösen sich von persönlichen Vorlieben für bestimmte Technologien.	Die Projektmitarbeiter wollen nur in der ihnen vertrauten Art unter Einsatz bestimmter Technologien entwickeln.	1	0 %	0 %	0 %	100 %
	Die Projektmitarbeiter sind bereit, sich in neue Systeme/Technologien einzuarbeiten.	Die Projektmitarbeiter sind nicht bereit sich in neue Systeme/Technologien einzuarbeiten.	1	100 %	100 %	0 %	0 %
	Die Projektmitarbeiter haben eine gesamtsystemorientierte Perspektive.	Die Projektmitarbeiter haben eine auf Anwendungssysteme fokussierte Perspektive.	1	100 %	0 %	0 %	100 %
Die Zusammenarbeit mit weiteren Unternehmen bei der Entwicklung der Integrationslösung	Die Realisierung der Integrationslösung wird intern durchgeführt.	Die Realisierung der Integrationslösung wird an externe Dienstleister vergeben.	1	100 %	100 %	0 %	0 %
	Die Entwicklung der Integrationslösung wird selbständig durchgeführt.	Die Entwicklung der Integrationslösung erfolgt gemeinsam mit weiteren externen Dienstleistern.	5	60 %	0 %	60 %	40 %
	Ein Teil der Verantwortung kann an einen erfahrenen externen Partner abgegeben werden.	Die Gesamtverantwortung liegt in der eigenen Hand.	1	0 %	0 %	100 %	0 %
	Die Entwicklung erfolgt in Zusammenarbeit mit einem Unternehmen, mit dem ein partnerschaftliches Verhältnis besteht.	Die Entwicklung erfolgt in Zusammenarbeit mit einem Unternehmen, mit dem ein kommerzielles Verhältnis besteht.	1	100 %	0 %	0 %	100 %
Die Qualität der allgemeinen Kommunikation unter den Projektbeteiligten	Die Schlüsselpersonen des Integrationsvorhabens haben eine gute Beziehung zueinander und kommunizieren viel.	Die Schlüsselpersonen des Integrationsvorhabens haben eine schlechte Beziehung zueinander und kommunizieren wenig.	1	100 %	0 %	100 %	0 %
	Die Kommunikation zwischen allen Projektbeteiligten ist gewährleistet.	Die einzelnen Projektbeteiligten arbeiten isoliert voneinander jeweils nur an ihren eigenen Arbeitspaketen.	1	0 %	100 %	0 %	0 %

Konstrukte zum Management des Integrationsvorhabens

Dem Themenbereich des Managements des Integrationsvorhabens wurden 4 Gruppen zugeordnet, die insgesamt 22 Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Qualität des Managements des Integrationsvorhabens	Es gibt ein starkes Projektmanagement.	Es gibt kein starkes Projektmanagement.	1	0 %	100 %	0 %	0 %
	Es gibt ein Projektmanagement mit klarem Scope und klar abgegrenzter Kompetenz.	Es gibt direkte Interventionen des Linienmanagements in das Projekt.	1	0 %	0 %	100 %	0 %
	Das Projektmanagement besitzt hohe soziale Kompetenz und fachliches Know-how.	Das Projektmanagement besitzt keine soziale Kompetenz und fachliches Know-how.	1	0 %	0 %	100 %	0 %
	Der Projektleiter ist sensibilisiert für Integrationsaspekte.	Der Projektleiter ist nicht sensibilisiert für Integrationsaspekte.	1	100 %	0 %	0 %	100 %
	Im Projekt ist ein Risikomanagement implementiert.	Im Projekt ist kein Risikomanagement implementiert.	1	0 %	0 %	100 %	0 %
	Es ist von Anfang an ein Changelmanagement aufgesetzt.	Es gibt kein Changelmanagement.	1	0 %	0 %	0 %	100 %
	Das Team der Projektmitarbeiter wird kontrolliert von oben nach unten aufgebaut.	Das Team der Projektmitarbeiter wird zu schnell aufgebaut.	1	0 %	0 %	0 %	100 %
	Die Planung des Projektes erfolgt anhand eines Vorgehensmodells.	Die Planung des Projektes erfolgt spontan und durch den Endtermin getrieben.	1	0 %	100 %	0 %	0 %
	Das Integrationsvorhaben wird in mehrere Meilensteine eingeteilt und diese Planung wird befolgt.	Das Integrationsvorhaben weist sehr lange Teilprojektphasen auf.	1	0 %	0 %	0 %	100 %
Das Ausmaß von vorausschauendem Vorgehen	Das Integrationsvorhaben wurde gut vorbereitet.	Das Integrationsvorhaben wurde unzureichend vorbereitet.	1	0 %	0 %	100 %	0 %
	Das Integrationsvorhaben folgt dem Grundsatz "erst denken, dann handeln".	In dem Integrationsvorhaben "sind Denken und Handeln eins".	1	0 %	100 %	0 %	0 %
Die Gestaltung der Aufbauorganisation des Integrationsvorhabens (Fortsetzung folgende Seite)	Zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter bestehen keine Interdependenzen.	Zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter bestehen starke Interdependenzen.	2	50 %	100 %	0 %	0 %
	Es gibt klare Organisationschnittstellen in der Projektorganisation.	Es gibt eine netzartige Kommunikationsstruktur in der Projektorganisation.	1	0 %	100 %	0 %	0 %
	Die Abstimmung und die Kommunikation zwischen Entwicklung und Test werden gefördert und durch dedizierte Projektmitarbeiter ("Integrationsteam") sichergestellt.	Die Abstimmung und die Kommunikation zwischen Entwicklung und Test werden nicht gefördert. Zwischen Entwicklung und Test besteht ein Nadelöhr.	1	100 %	100 %	0 %	0 %
	Die Projektorganisation ist teamorientiert.	Die Projektorganisation ist hierarchisch.	1	0 %	100 %	0 %	0 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die Gestaltung der Aufbauorganisation des Integrationsvorhabens	Es gibt klar definierte Verantwortlichkeiten hinsichtlich der Integrationsfunktionalität.	Es gibt eine Vermischung der Verantwortlichkeiten hinsichtlich der Integrationsfunktionalität und fachlicher Funktionalität.	1	0 %	100 %	0 %	0 %
(Fortsetzung)	Die Verantwortlichkeiten der Projektbeteiligten sind klar definiert. Die organisatorische Gestaltung des Integrationsvorhabens ist projektorientiert, bei Ausrichtung an den zu liefernden Ergebnissen.	Die Verantwortlichkeiten der Projektbeteiligten sind unklar. Das Integrationsvorhaben wird aus der Linienorganisation des Unternehmens heraus durchgeführt. Die Ergebnisse der einzelnen Linien müssen kombiniert werden.	1	0 %	0 %	100 %	0 %
	Es gibt ein zentrales Mandat für die Modellierung der zu unterstützenden Geschäftsprozesse.	Das Mandat für die Modellierung der zu unterstützenden Geschäftsprozesse ist auf mehrere Teilprojekte verteilt.	1	100 %	0 %	0 %	100 %
Der verfolgte Entwicklungsansatz	Der Kunde will zu einem bestimmten Termin einen definierten Lieferumfang haben. Es wird ein sequenzieller/dokumentenbasierter Entwicklungsansatz verfolgt.	Der Kunde will zu einem bestimmten Termin einen definierten Lieferumfang haben. Es wird ein evolutionärer Entwicklungsansatz verfolgt.	1	0 %	0 %	100 %	0 %
	Der Kunde ist terminorientiert und in der Gestaltung der Funktionalität flexibler. Es wird ein evolutionärer Entwicklungsansatz verfolgt.	Der Kunde ist terminorientiert und in der Gestaltung der Funktionalität flexibler. Es wird ein sequenzieller/dokumentenbasierter Entwicklungsansatz verfolgt.	1	0 %	0 %	100 %	0 %
	Unter den Projektbeteiligten herrscht ein Konsens über die verfolgte Entwicklungsmethode.	Unter den Projektbeteiligten herrscht kein Minimalkonsens über die verfolgte Entwicklungsmethode.	1	100 %	0 %	100 %	0 %
	Die Releasezyklen der zu integrierenden Anwendungssysteme und die Vorgehensmodelle, auf denen ihre Entwicklung beruht, können im Integrationsvorhaben berücksichtigt werden.	Die zu integrierenden Anwendungssysteme besitzen unterschiedliche Releasezyklen und ihre Entwicklung beruht auf unterschiedlichen Vorgehensmodellen.	1	100 %	0 %	100 %	0 %

Konstrukte zu den Rahmenbedingungen des Integrationsvorhabens

Der letzte der abgegrenzten Themenbereiche bezieht sich auf die Rahmenbedingungen des Integrationsvorhabens. Ihm wurden 10 Gruppen zugeordnet, die insgesamt 37 Konstrukte umfassen.

Gruppe	Persönliche Konstrukte		Häu- figkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Cha- rak- teristisch	Bedeutung für den Ent- wicklungsaufwand		
					+	o	-
Die Dauer des Integrations- vorhabens	Das Integrationsvorhaben hat eine kurze Laufzeit.	Das Integrationsvorhaben hat eine lange Laufzeit.	2	0 %	50 %	0 %	50 %
	Das Integrationsvorhaben umfasst wenige Releases.	Das Integrationsvorhaben umfasst viele Releases.	2	50 %	100 %	0 %	0 %
Die Neuartigkeit des Integrations- vorhabens	Das Integrationsvorhaben stellt die Fortführung eines vorange- henden Projektes dar.	Das Integrationsvorhaben ist vollständig neu.	1	100 %	0 %	0 %	100 %
Die Klarheit der Zielsetzung des Integrations- vorhabens	Das Integrationsvorhaben ist auf die Bedürfnisse des Einzelfalls ausgerichtet.	Das Integrationsvorhaben soll als Vorbild/Vorreiter für ein veränder- tes Vorgehen in der IT dienen.	1	0 %	0 %	0 %	100 %
	Das Integrationsvorhaben verfolgt klar abgegrenzte Zielvorgaben.	Das Integrationsvorhaben stellt einen ersten Schritt auf einem langen Weg dar.	1	100 %	100 %	0 %	0 %
	Die Entwicklung einer umfassen- den Integrationsarchitektur wurde explizit als Infrastrukturprojekt aufgesetzt.	Das Vorhaben, eine umfassende Integrationsarchitektur zu entwi- ckeln, ist historisch gewachsen.	1	100 %	100 %	0 %	0 %
	Neben der Anwendungsintegrati- on sind wenige begleitenden Maßnahmen durchzuführen.	Neben der Anwendungsintegration sind viele begleitenden Maßnah- men durchzuführen.	1	0 %	0 %	100 %	0 %
	Mit der Anwendungsintegration werden Kostenvorteile auf hoher Budgetebene angestrebt.	Das Integrationsvorhaben ist charakterisiert durch ein perman- tes Streben nach Kostenvorteilen auch im Kleinen.	1	0 %	0 %	0 %	100 %
Die Bedeutung des Integrationsvorha- bens für den Kun- den	Die Anwendungsintegration hat lediglich interne Auswirkungen für die beteiligten Unternehmen.	Die Anwendungsintegration hat eine Außenwirkung im Hinblick auf den Endkunden.	2	0 %	50 %	50 %	0 %
	Ein Misserfolg des Integrations- vorhabens ist für den Kunden nicht geschäftsgefährdent.	Das Integrationsvorhaben ist erfolgskritisch für den Kunden.	1	0 %	0 %	0 %	100 %
	Die zu integrierenden Anwen- dungssysteme sind nicht ge- schäftskritisch.	Die zu integrierenden Anwen- dungssysteme sind geschäftskri- tisch.	1	0 %	100 %	0 %	0 %
Das Ausmaß der Internationalität des Integrations- vorhabens	Das Integrationsvorhaben ist national.	Das Integrationsvorhaben ist international.	4	50 %	0 %	75 %	25 %
	Die Dokumentation der Integrati- onslösung erfolgt in einer Spra- che.	Die Dokumentation der Integrati- onslösung erfolgt in mehreren Sprachen.	1	0 %	0 %	0 %	100 %
	Die Projektbeteiligten sind mit der Projektsprache sehr vertraut.	Die Projektbeteiligten sind mit der Projektsprache nicht vertraut.	2	0 %	0 %	0 %	100 %
	Die Projektbeteiligten entstam- men einem gemeinsamen Sprach- und Kulturraum.	Die Projektbeteiligten entstammen unterschiedlichen Sprach- und Kulturräumen.	4	0 %	0 %	75 %	25 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die räumliche Verteilung der Projektbeteiligten	Die Projektbeteiligten befinden sich an einem Standort.	Die Projektbeteiligten sind räumlich getrennt.	8	25 %	25 %	50 %	25 %
	Die Projektbeteiligten befinden sich in derselben Zeitzone.	Die Projektbeteiligten befinden sich in unterschiedlichen Zeitzonen.	2	0 %	0 %	100 %	0 %
Die verfügbare Infrastruktur zur Durchführung des Integrationsvorhabens	Es gibt angemessene getrennte Umgebungen für Entwicklung, Test und Abnahme.	Die Entwicklung der Integrationslösung erfolgt unmittelbar in der Produktivumgebung.	2	50 %	0 %	50 %	50 %
	Die Produktivumgebung kann in Entwicklungs- und Testumgebungen nachgebildet werden.	Die Produktivumgebung kann nicht nachgebildet werden. Entwicklungs- und Testumgebungen entsprechen nicht der Produktivumgebung.	1	100 %	0 %	0 %	100 %
	Die Integrationslösung muss lediglich in der Produktivumgebung eingeführt ("deployed") werden.	Die Integrationslösung muss in Entwicklungs-, Test- und Produktivumgebungen eingeführt ("deployed") werden.	1	0 %	0 %	0 %	100 %
	Die räumliche und technische Ausstattung der Projektmitarbeiter ist optimal.	Es ist keine angemessene räumliche und technische Infrastruktur für die Projektmitarbeiter vorhanden.	1	0 %	0 %	0 %	100 %
Die Abhängigkeit des Integrationsvorhabens von anderen Projekten	Bei dem Integrationsvorhaben handelt es sich um ein eigenständiges Projekt.	Das Integrationsvorhaben ist ein Teilaspekt eines größeren, umfassenden Projektes.	1	0 %	0 %	0 %	100 %
	Es gibt keine Parallelprojekte von denen man abhängig ist.	Es gibt Parallelprojekte von denen man abhängig ist.	1	0 %	100 %	0 %	0 %
	Es ist keine Abstimmung mit anderen Projekten erforderlich.	Es ist eine Abstimmung mit anderen Projekten erforderlich.	1	0 %	0 %	0 %	100 %
	Man kann die Releaseplanung für das Integrationsvorhaben eigenständig vornehmen.	Das Integrationsvorhaben muss in eine bestehende Releaseplanung eingebettet werden.	1	100 %	0 %	100 %	0 %
	Das Integrationsvorhaben ist ein Teilaspekt eines Projektes mit kurzer Gesamtdauer bzw. geringem Gesamtvolumen.	Das Integrationsvorhaben ist ein Teilaspekt eines Projektes mit langer Gesamtdauer bzw. großem Gesamtvolumen.	1	0 %	0 %	0 %	100 %
(Fortsetzung folgende Seite)	Eine beim Kunden bereits eingesetzte Integrationsplattform ist für die Integrationslösung vorgegeben.	Man hat Wahlfreiheit bezüglich eines neu einzuführenden Integrationsproduktes.	1	0 %	0 %	100 %	0 %
	Man hat Wahlfreiheit bezüglich der einzusetzenden Integrationsprodukte und der verwendeten Schnittstellen.	Es gibt Vorgaben bezüglich der einzusetzenden Integrationsprodukte oder Schnittstellen, die für die Lösung des Integrationsproblems nicht geeignet oder notwendig sind.	3	33 %	0 %	67 %	33 %
	Es gibt klare Vorgaben des Kunden für die Entwicklung ("starke Governance" beim Kunden).	Die Entwicklung wird nicht durch den Kunden kontrolliert ("keine starke Governance" beim Kunden).	2	50 %	50 %	50 %	0 %
	Es gibt keine bestimmten Anforderungen des Kunden, wie eine Software zu entwickeln ist.	Es gibt bestimmte Anforderungen des Kunden, wie eine Software zu entwickeln ist.	1	0 %	0 %	0 %	100 %

Gruppe	Persönliche Konstrukte		Häufigkeit	Anteil der Beurteilungen			
	Weniger Aufwand wenn...	Höherer Aufwand wenn...		Charakteristisch	Bedeutung für den Entwicklungsaufwand		
					+	o	-
Die bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben (Fortsetzung)	Der Kunde mischt sich nicht in die Entwicklung ein.	Der Kunde mischt sich in die Entwicklung ein.	2	0 %	50 %	0 %	50 %
	Das Integrationsvorhaben bezweckt die einmalige Verbindung der zu integrierenden Anwendungssysteme.	Der Lebenszyklus der zu integrierenden Anwendungssysteme muss bei der Entwicklung der Integrationslösung berücksichtigt werden. Systematische Veränderungen der zu integrierenden Anwendungssysteme (insbesondere der Schnittstellen) müssen unterstützt werden	1	0 %	0 %	0 %	100 %
	Die Integrationslösung wird aufgrund eigener Konzepte maßgeschneidert.	Die Integrationslösung basiert auf einem standardisierten Paradigma, dessen Umsetzung einen Overhead bedeutet.	2	50 %	0 %	50 %	50 %
	Datenschutzrechtliche Bestimmungen sind von geringer Bedeutung.	Datenschutzrechtliche Bestimmungen sind von großer Bedeutung.	1	100 %	0 %	0 %	100 %
Das verfügbare Budget / die zulässige Entwicklungsdauer	Das Budget des Kunden ist knapp. Der Kunde hat beschränkte finanzielle Mittel.	Das Budget des Kunden ist ausreichend. Der Kunde hat große finanzielle Mittel.	1	0 %	0 %	100 %	0 %
	Man kann bedächtig entwickeln.	Es herrscht großer Zeitdruck.	1	0 %	100 %	0 %	0 %
	Es gibt zwar einen vorgegebenen Plan für den Projektverlauf aber man kann Abweichungen von den Meilensteinen einkalkulieren.	Es gibt vorgegebene Releasetermine, die nicht verändert werden können.	1	100 %	0 %	100 %	0 %

Anhang B: Eine Checkliste für die Aufwandschätzung von Integrationsvorhaben

**Eine Checkliste zur Unterstützung der Aufwandschätzung
von Entwicklungsvorhaben zur Integration betrieblicher Anwendungssysteme**

Holger Wagner

März 2006

Lehrstuhl für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln,
Albertus-Magnus-Platz, 50923 Köln, Tel. (0221) 470-5368, Fax (0221) 470-5386

Wie viel Aufwand verursacht die Integration betrieblicher Anwendungssysteme? Viele werden sagen: „Das kommt darauf an.“ Aber worauf?

In diesem Dokument wird eine Liste von Einflussfaktoren auf den Entwicklungsaufwand der Integration betrieblicher Anwendungssysteme aufgeführt. Auf diese Weise sollen Anregungen für die Erstellung und die Beurteilung von Schätzergebnissen gegeben werden.

Insbesondere kann Ihnen die Liste dabei helfen, wichtige Aufwandstreiber in Erinnerung zu rufen. Darüber hinaus kann Sie die Liste dabei unterstützen, die einem Schätzergebnis zugrunde liegenden Informationen und Annahmen zu dokumentieren. Das Schätzergebnis kann somit nachvollzogen und differenziert dem tatsächlich entstandenen Entwicklungsaufwand gegenübergestellt werden. Auf diese Weise kann diese Liste einen Beitrag zu einer kontinuierlichen Verbesserung der Schätzgenauigkeit leisten.

Die in der Liste aufgeführten Einflussfaktoren geben die umfangreichen Erfahrungen und Ansichten von 22 kompetenten Praktikern wieder, die in einer Studie des Lehrstuhls für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln im Zeitraum von August bis November 2005 erhoben wurden.

Sie beziehen sich auf Softwareentwicklungsvorhaben, in denen betriebliche Anwendungssysteme eines oder mehrerer Unternehmen verknüpft werden, um automatisierte Interaktionen der Anwendungssysteme zur systemübergreifenden Unterstützung von Geschäftsprozessen zu realisieren (kurz: Integrationsvorhaben). Die Interaktionen zwischen den integrierten Anwendungssystemen können den Austausch von Informationen, die Erschließung der Funktionalität eines Anwendungssystems oder die Abwicklung mehrstufiger Prozesse betreffen. Sie werden i. d. R. über zwischengeschaltete Softwareschichten (Middleware) realisiert.

Die Liste ist in neun Themenbereiche untergliedert:

- | | |
|--|---|
| 1. die Anforderungen an die Integrationslösung | 5. das Testen der Integrationslösung und die Überleitung in die Nutzung |
| 2. die Charakteristika der zu integrierenden Anwendungssysteme | 6. der Kunde / die Stakeholder der Integrationslösung |
| 3. die Gestaltung der Integrationslösung | 7. die Projektmitarbeiter |
| 4. die eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge | 8. das Management des Integrationsvorhabens |
| | 9. die Rahmenbedingungen des Integrationsvorhabens |

Innerhalb der Themenbereiche werden jeweils verschiedene Merkmale von Integrationsvorhaben aufgeführt. Die möglichen Ausprägungen der Merkmale sind daraufhin im Wortlaut der Praktiker illustriert. Dabei werden jeweils zwei gegensätzliche Ausprägungen beschrieben, die stets in gleicher Weise angeordnet sind: erfahrungsgemäß ist die linke Ausprägung mit einem geringeren und die rechte Ausprägung mit einem höheren Entwicklungsaufwand verbunden. Zwischen ihnen sind jeweils fünfstufige Skalen abgebildet, anhand derer Sie das von Ihnen betrachtete Integrationsvorhaben einordnen können.

Sollten Sie Interesse an weiterführenden Informationen zu dieser Checkliste und der ihr zugrunde liegenden Studie haben, können Sie sich sehr gerne an den Lehrstuhl für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln wenden.

E-Mail: forschung@systementwicklung.uni-koeln.de

Tel: 0 22 1 / 470 - 53 68

1. Die Anforderungen an die Integrationslösung

☐ Der allgemeine Umfang der Aufgabenstellung

Der Integrationsumfang ist gering.



Der Integrationsumfang ist groß.

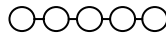
Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht sind gering.



Umfang und Komplexität der Aufgabenstellung aus fachlicher Sicht ist groß.

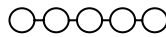
☐ Die Anzahl, Komplexität die Art und die der zu unterstützenden Geschäftsprozesse

Die Anzahl der zu unterstützenden Geschäftsprozesse ist gering.



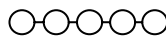
Die Anzahl der zu unterstützenden Geschäftsprozesse ist groß.

Es gilt Geschäftsprozesse einer Art ("Prozessfamilie") zu unterstützen.



Es gilt Geschäftsprozesse vieler unterschiedlicher Arten ("Prozessfamilien") zu unterstützen.

Die Komplexität eines zu unterstützenden Geschäftsprozesses ist gering.



Die Komplexität eines zu unterstützenden Geschäftsprozesses ist groß.

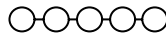
☐ Die Anzahl und die Komplexität der zu unterstützenden Use Cases

Die Anzahl der zu unterstützenden Use Cases ist gering.



Die Anzahl der zu unterstützenden Use Cases ist groß.

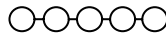
Die Komplexität eines zu unterstützenden Use Cases ist gering.



Die Komplexität eines zu unterstützenden Use Cases ist groß.

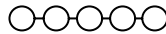
☐ Die mit der Anwendungsintegration verfolgte Perspektive

Die Anwendungsintegration betrifft ausschließlich technische Aspekte.



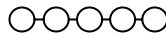
Die Anwendungsintegration betrifft technische und fachliche Aspekte.

Die Anwendungssysteme werden integriert, um damit Benutzer zu bedienen. Die Ansprüche an die Anwendungsintegration werden somit durch Menschen gestellt.



Die Anwendungssysteme werden integriert, um damit weitere Softwaresysteme zu bedienen. Die Ansprüche an die Anwendungsintegration werden somit durch Softwaresysteme gestellt.

Die Integrationslösung basiert auf der Daten-Ebene.



Die Integrationslösung umfasst auch Interaktionen mit Benutzern und macht damit insbesondere die Unterstützung von Berechtigungskonzepten und Workflows erforderlich.

Die Anwendungsintegration ist informationsorientiert.



Die Anwendungsintegration ist prozessorientiert.

Die Interaktionen der zu integrierenden Anwendungssysteme erfolgen vollständig automatisiert.



Während der Interaktionen der zu integrierenden Anwendungssysteme sind manuelle Interventionen erforderlich.

Die zu unterstützenden Geschäftsprozesse werden nicht vollständig automatisiert.



Die zu unterstützenden Geschäftsprozesse werden vollständig automatisiert.

☐ Die Neuartigkeit der Integrationslösung

Die zu entwickelnde Integrationslösung baut auf einer bereits bestehenden Integrationslösung auf.



Die zu entwickelnde Integrationslösung wird vollständig neu aufgebaut.

Eine bereits bestehende Integrationslösung soll nachgebaut und ersetzt werden.



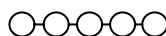
Bislang ist keine Integrationslösung vorhanden.

Die Integrationslösung basiert auf einer etablierten und verifizierten Architektur.



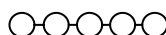
Die Integrationslösung basiert auf einer nicht ausreichend erprobten Architektur.

Eine bereits zentralisierte Anwendungslandschaft soll durch eine zentrale Lösung integriert werden. Es muss somit eine bestehende IT-Architektur umstrukturiert werden.



Eine verteilte Anwendungslandschaft soll durch die Anwendungsintegration zentralisiert werden. Es muss somit grundsätzlich eine andere IT-Architektur geschaffen werden.

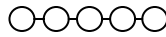
Es ist nicht erforderlich, fachliche Funktionalität neu zu entwickeln.



Es ist erforderlich, fachliche Funktionalität neu zu entwickeln.

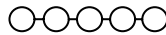
☐ Die Neuartigkeit der durch die Integrationslösung zu unterstützenden Geschäftsprozesse

Die zu unterstützenden Geschäftsprozesse sind etabliert.



Die zu unterstützenden Geschäftsprozesse müssen erst definiert werden.

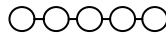
Ein durch die Integrationslösung unterstützter Geschäftsprozess ist ein Ersatz für einen einheitlichen bestehenden Geschäftsprozess.



Ein durch die Integrationslösung unterstützter Geschäftsprozess ist ein Ersatz für mehrere unterschiedliche Geschäftsprozesse.

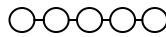
☐ Die Klarheit der Anforderungen

Der Kunde kennt und versteht das Integrationsproblem.



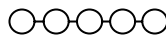
Der Kunde muss erst eine Integrationslösung sehen und verstehen, um das Integrationsproblem genau zu verstehen.

Es gibt eine klare Aufgabenstellung.



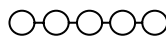
Der Kunde ist sich über seine Ziele unklar. Die Aufgabenstellung entwickelt sich nach und nach.

Die zu unterstützenden Geschäftsprozesse sind klar definiert.



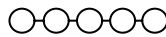
Die zu unterstützenden Geschäftsprozesse sind nicht klar definiert.

Die fachlichen Interdependenzen zwischen den zu integrierenden Anwendungssystemen können genau definiert werden.



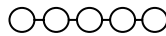
Die fachlichen Interdependenzen zwischen den zu integrierenden Anwendungssystemen können nicht definiert werden.

Es erfolgt keine Experimentierung im Laufe des Integrationsvorhabens.



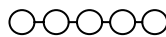
Es erfolgt viel Experimentierung im Laufe des Integrationsvorhabens.

Die Anforderungen sind klar definiert und dokumentiert.



Die Anforderungen sind nicht klar definiert und dokumentiert.

Die Dokumentation der Anforderungen basiert auf formaler Modellierung. Es gibt nur geringen Interpretationsspielraum bezüglich des Analyseergebnisses.



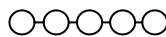
Die Dokumentation der Anforderungen ist lediglich verbalsprachlich. Das Analyseergebnis ist frei interpretierbar.

Die Projektmitarbeiter wissen genau was sie machen müssen.



Die Projektmitarbeiter füllen Lücken in ihrem Verständnis mit eigenen Entscheidungen.

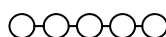
Es gibt eine inhaltliche Interpretation von Qualität. Die Anforderungen an die Qualität der Integrationslösung sind gestaltbar.



Es gibt eine formale Interpretation von Qualität. Die Anforderungen an die Qualität der Integrationslösung sind nicht inhaltlich sondern quantitativ festgelegt (z. B. dürfen bei der Abnahme maximal x schwere Fehler auftreten).

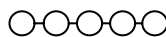
☐ Die Änderungsdynamik der Anforderungen

Die Anforderungen sind klar definiert und stabil.



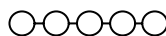
Im Laufe des Integrationsvorhabens treten neue Anforderungen auf.

Es gibt eine klare Aufgabenstellung.



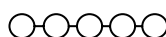
Das Integrationsvorhaben unterliegt einem starken Einfluss von Außen. Es findet in einem dynamischen Umfeld statt.

Das Integrationsvorhaben erfolgt in einem stabilen und bekannten Systemumfeld.



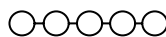
Das Integrationsvorhaben erfolgt in einem dynamischen Systemumfeld. Das Systemumfeld ändert sich im Lauf des Integrationsvorhabens.

Der Projektumfang ändert sich während der Projektlaufzeit nicht.



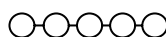
Der Projektumfang ändert sich während der Projektlaufzeit.

Die Organisation des Kunden ändert sich nicht im Laufe des Integrationsvorhabens.



Die Organisation des Kunden ändert sich im Laufe des Integrationsvorhabens.

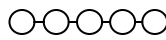
Es gibt ein etabliertes Change Request System.



Die Verwaltung der Change Requests muss neu erarbeitet und umgesetzt werden.

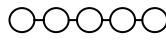
☐ **Die Anforderungen an die qualitätsbezogenen Merkmale der Integrationslösung**

Die qualitätsbezogenen/nicht-funktionalen Anforderungen an die Integrationslösung sind gering.



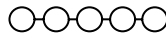
Die qualitätsbezogenen/nicht-funktionalen Anforderungen an die Integrationslösung sind hoch.

Die Sicherheitsanforderungen an die Integrationslösung sind gering.



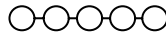
Die Sicherheitsanforderungen an die Integrationslösung sind hoch.

Es ist nicht kritisch, dass über die zu integrierenden Anwendungssysteme hinweg Transaktionssicherheit besteht.



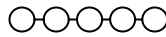
Transaktionssicherheit über die zu integrierenden Anwendungssysteme hinweg ist erforderlich.

Die Anforderungen an die Verfügbarkeit der Integrationslösung sind gering.



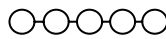
Die Anforderungen an die Verfügbarkeit der Integrationslösung sind hoch.

Die Anforderungen an die Ausfallsicherheit der Integrationslösung sind gering.



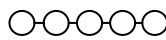
Die Anforderungen an die Ausfallsicherheit der Integrationslösung sind hoch.

Die Anforderungen an die Performanz der Integrationslösung sind gering.



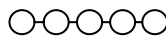
Die Anforderungen an die Performanz der Integrationslösung sind hoch.

Die Anforderungen an die Skalierbarkeit der Integrationslösung sind gering.



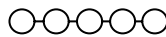
Die Anforderungen an die Skalierbarkeit der Integrationslösung sind hoch.

Die Anforderungen an die Revisionssicherheit der Integrationslösung sind gering.



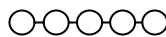
Die Anforderungen an die Revisionssicherheit der Integrationslösung sind hoch.

Die Integrationslösung besitzt keine komplexen Fehlerbehandlungsmechanismen. Fehler werden nur protokolliert.



Die Integrationslösung besitzt komplexe Fehlerbehandlungsmechanismen.

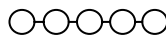
Bei der Gestaltung der Integrationslösung werden nur unmittelbare Ziele verfolgt.



Bei der Gestaltung der Integrationslösung werden eine hohe Flexibilität und Wiederverwendbarkeit angestrebt.

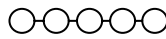
☐ **Die Anforderungen an die Protokollierung der Interaktionen zwischen den zu integrierenden Anwendungssystemen**

Bei der Interaktion von Anwendungssystemen besteht kein fachlicher Überwachungsbedarf ("track and trace").



Bei der Interaktion von Anwendungssystemen besteht großer fachlicher Überwachungsbedarf ("track and trace").

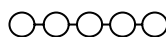
Für die Interaktionen der zu integrierenden Anwendungssysteme ist lediglich eine eingeschränkte Kontakthistorie erforderlich.



Für die Interaktionen der zu integrierenden Anwendungssysteme ist eine umfangreiche Kontakthistorie erforderlich.

☐ **Die Anzahl der Benutzer und Benutzerrollen**

Die Anzahl der Benutzer ist gering.



Die Anzahl der Benutzer ist groß.

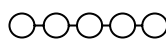
Die Anzahl der Benutzerrollen ist gering.



Die Anzahl der Benutzerrollen ist groß.

☐ **Die Verteilung der Nutzung der Integrationslösung**

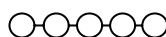
Die Integrationslösung wird zentral an einem Standort genutzt.



Die Integrationslösung wird dezentral an verschiedenen Standorten genutzt.

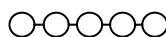
☐ **Die Notwendigkeit des Abgleichs der Stammdaten der zu integrierenden Anwendungssysteme**

Es ist keine initiale Stammdatensynchronisation erforderlich.



Es ist eine initiale Stammdatensynchronisation erforderlich.

Es ist keine Stammdatensynchronisation zur Laufzeit notwendig.



Es ist eine Stammdatensynchronisation zur Laufzeit erforderlich.

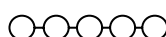
Es gibt einen festen und durchgehenden Satz von Stammdaten.



Die Stammdaten weisen eine hohe Dynamik auf.

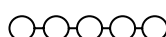
☐ **Die Notwendigkeit einer Migration**

Es ist keine Migration von Daten erforderlich.



Es ist eine Migration von Daten erforderlich.

Ein Parallelbetrieb zu migrierender Systeme ist nicht erforderlich.

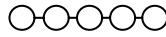


Ein Parallelbetrieb zu migrierender Systeme ist erforderlich.

2. Die Charakteristika der zu integrierenden Anwendungssysteme

☐ Die Anzahl und die Komplexität der zu integrierenden Anwendungssysteme

Die Anzahl der zu integrierenden Anwendungssysteme ist gering.



Die Anzahl der zu integrierenden Anwendungssysteme ist groß.

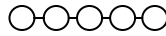
Die Komplexität eines zu integrierenden Anwendungssystems ist gering.



Die Komplexität eines zu integrierenden Anwendungssystems ist groß.

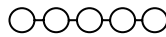
☐ Die Arten zu integrierender Anwendungssysteme

Im Rahmen der Integrationslösung werden keine Telekommunikationssysteme angebunden.



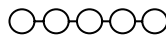
Im Rahmen der Integrationslösung werden Telekommunikationssysteme angebunden.

Ein zu integrierendes Anwendungssystem ist in räumlicher Hinsicht statisch.



Ein zu integrierendes Anwendungssystem ist in räumlicher Hinsicht mobil (z. B. im Zusammenhang mit Telemetrie).

Eine Interaktion findet zwischen zwei Server-Systemen statt.



Eine Interaktion findet zwischen Server- und Client-Systemen statt.

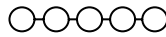
Alle zu integrierenden Anwendungssysteme sind Standardsoftwaresysteme.



Alle zu integrierenden Anwendungssysteme sind Individualsoftwaresysteme.

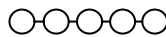
☐ Die Verfügbarkeit der Schnittstellen der zu integrierenden Anwendungssysteme

Ein zu integrierendes Anwendungssystem hat wohldefinierte Schnittstellen.



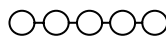
Ein zu integrierendes Anwendungssystem hat keine wohldefinierten Schnittstellen.

Die Reife einer Schnittstelle eines Anwendungssystems ist hoch.



Eine Schnittstelle eines Anwendungssystems muss neu entwickelt werden.

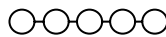
Der Kunde hat vollständige Nutzungsmöglichkeiten und -rechte der Schnittstellen eines zu integrierenden Anwendungssystems sowie der zugehörigen Dokumentation.



Der Kunde hat keine Nutzungsmöglichkeiten und -rechte der Schnittstellen eines zu integrierenden Anwendungssystems sowie der zugehörigen Dokumentation.

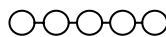
☐ Die Eignung der Schnittstellen der zu integrierenden Anwendungssysteme für die Integrationslösung

Ein zu integrierendes Anwendungssystem ist technisch für die Integration vorbereitet bzw. geeignet.



Ein zu integrierendes Anwendungssystem ist technisch nicht für die Integration vorbereitet bzw. geeignet.

Ein zu integrierendes Anwendungssystem besitzt eine geeignete Schnittstelle für eine erforderliche Online-Verarbeitung.



Eine Online-Verarbeitung muss unter Verwendung einer Batch-Schnittstelle des zu integrierenden Anwendungssystems realisiert werden.

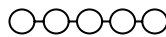
☐ Die Qualität der Schnittstellen der zu integrierenden Anwendungssysteme

Die Qualität einer verwendeten Schnittstelle ist hoch.



Die Qualität einer verwendeten Schnittstelle ist gering.

Die Qualität der von einem zu integrierenden Anwendungssystem gelieferten Daten ist hoch.



Die Qualität der von einem zu integrierenden Anwendungssystem gelieferten Daten ist niedrig.

☐ Die Standardbasiertheit der Schnittstellen der zu integrierenden Anwendungssysteme

Eine Schnittstelle eines Anwendungssystems basiert auf Standards.



Eine Schnittstelle eines Anwendungssystems ist proprietär.

☐ Die Änderungsdynamik der Schnittstellen der zu integrierenden Anwendungssysteme

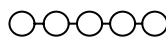
Die verwendeten Schnittstellen sind konstant.



Die verwendeten Schnittstellen unterliegen einer hohen Änderungshäufigkeit.

☐ Die Notwendigkeit der Anpassung der zu integrierenden Anwendungssysteme

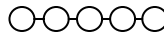
Ein zu integrierendes Anwendungssystem muss nicht verändert werden.



Ein zu integrierendes Anwendungssystem muss angepasst werden.

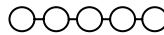
☐ Die Heterogenität der zu integrierenden Anwendungssysteme

Die zu integrierenden Anwendungssysteme basieren auf den gleichen Technologien.



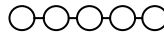
Die zu integrierenden Anwendungssysteme basieren auf unterschiedlichen Technologien.

Die zu integrierenden Anwendungssysteme sind hinsichtlich ihrer technischen und konzeptionellen Charakteristika einheitlich.



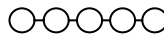
Die zu integrierenden Anwendungssysteme sind hinsichtlich ihrer technischen und konzeptionellen Charakteristika unterschiedlich.

Die zu integrierenden Anwendungssysteme basieren auf einem einheitlichen Datenmodell.



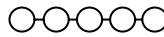
Die zu integrierenden Anwendungssysteme basieren auf unterschiedlichen Datenmodellen.

Es gibt ein homogenes Berechtigungskonzept über die zu integrierenden Anwendungssysteme hinweg.



Es gibt unterschiedliche Berechtigungskonzepte für die zu integrierenden Anwendungssysteme.

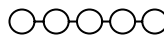
Die Schnittstellen der zu integrierenden Anwendungssysteme sind gleichartig.



Die Schnittstellen der zu integrierenden Anwendungssysteme sind unterschiedlich.

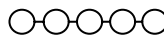
☐ Die Systemumgebung der zu integrierenden Anwendungssysteme

Ein zu integrierendes Anwendungssystem wird auf einer modernen Plattform betrieben.



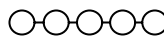
Ein zu integrierendes Anwendungssystem wird auf einer unmodernen Plattform betrieben.

Die zu integrierenden Anwendungssysteme sind unabhängig.



Die zu integrierenden Anwendungssysteme sind in eine komplexe gewachsene Anwendungslandschaft eingebunden.

Die Anwendungsintegration betrifft lediglich die Verknüpfung von Servern im Rechenzentrum-Umfeld.



Die Anwendungsintegration erfolgt in einer komplexen Netzwerkumwelt. Die Anzahl der zu berücksichtigenden technischen Komponenten ist hoch.

Die zu integrierenden Anwendungssysteme befinden sich in einer einheitlichen Systemumgebung.



Die zu integrierenden Anwendungssysteme befinden sich in unterschiedlichen Systemumgebungen.

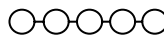
Das Umfeld der zu integrierenden Anwendungssysteme ist einheitlich oder muss bei der Integration nicht berücksichtigt werden.



Die zu integrierenden Anwendungssysteme besitzen heterogene Umfelder, die bei der Integration berücksichtigt werden müssen.

☐ Die Qualität der Dokumentation und der Ansprechpartner im Hinblick auf die zu integrierenden Anwendungssysteme

Die verfügbare Dokumentation ist quantitativ und qualitativ hinreichend.



Es ist keine technische oder fachliche Dokumentation verfügbar.

Die vorgefundene Anwendungslandschaft ist definiert und dokumentiert.



Die vorgefundene Anwendungslandschaft ist nicht definiert und dokumentiert.

Eine Schnittstelle eines Anwendungssystems ist sehr gut dokumentiert.



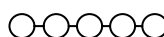
Eine Schnittstelle eines Anwendungssystems ist nicht dokumentiert.

Es ist möglich, Personen zu finden, welche die zu integrierenden Anwendungssysteme genau kennen.



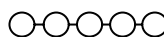
Es ist nicht möglich, Personen zu finden, welche die zu integrierenden Anwendungssysteme genau kennen.

Es gibt klare vertragliche Regelungen mit dem Hersteller eines zu integrierenden Anwendungssystems.



Ein zu integrierendes Anwendungssystem ist eine alte Individualsoftware. Es gibt keinen klaren Ansprechpartner und kein Know-how für das Anwendungssystem.

Es gibt eine gute Zusammenarbeit mit dem Hersteller eines zu integrierenden Anwendungssystems.



Der Hersteller eines zu integrierenden Anwendungssystems unterstützt das Integrationsvorhaben nicht.

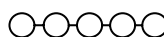
☐ Der Betrieb der zu integrierenden Anwendungssysteme

Die zu integrierenden Anwendungssysteme werden selbst betrieben.



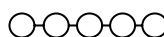
Der Betrieb der zu integrierenden Anwendungssysteme ist an externe Dienstleister ausgelagert

Die zu integrierenden Anwendungssysteme sind innerhalb der Organisation.



Die zu integrierenden Anwendungssysteme sind außerhalb der Organisation.

Es herrscht Informationstransparenz im Hinblick auf Betrieb und Nutzung eines zu integrierenden Anwendungssystems.

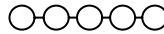


Es besteht ein Informationsdefizit im Hinblick auf Betrieb und Nutzung eines zu integrierenden Anwendungssystems.

3. Die Gestaltung der Integrationslösung

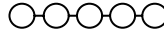
☐ Die Investition in Entwurf und Architektur der Integrationslösung

Die Gruppe der Projektmitarbeiter, die mit der Entwicklung der Architektur betraut sind, ist groß und spielt eine wichtige Rolle im Projekt.



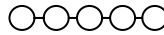
Die Gruppe der Projektmitarbeiter, die mit der Entwicklung der Architektur betraut sind, ist klein und hat lediglich eine beratende Funktion.

Bei der Entwicklung wird eine architektonische Durchgängigkeit sichergestellt.



Es werden lediglich rudimentäre architektonische Vorgaben gemacht.

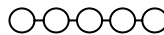
Fachliches Design, technisches Design und Schnittstellendesign der Integrationslösung werden zentral erstellt und konsequent umgesetzt.



Fachliches Design, technisches Design und Schnittstellendesign der Integrationslösung werden nicht zentral erstellt oder nicht eingehalten.

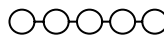
☐ Die Komplexität der Integrationsarchitektur

Die Komplexität der Integrationsarchitektur ist niedrig.



Die Komplexität der Integrationsarchitektur ist groß.

Die Integrationsarchitektur besteht aus wenigen Komponenten.



Die Integrationsarchitektur besteht aus vielen Komponenten.

Die Integrationsarchitektur besteht aus einer Schicht ("one tier architecture").



Die Integrationsarchitektur besteht aus mehreren Schichten ("multi tier architecture").

☐ Die Änderungsdynamik der Integrationsarchitektur

Die Integrationsarchitektur ist klar geplant.



Die Integrationsarchitektur ist im Fluss.

Es gibt keine Änderung der Integrationsarchitektur während der Projektlaufzeit.



Die Integrationsarchitektur ändert sich während der Projektlaufzeit.

☐ Die Struktur der Kommunikationsbeziehungen zwischen den zu integrierenden Anwendungssystemen

Zwischen den zu integrierenden Anwendungssystemen besteht eine 1-zu-1-Beziehung.



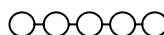
Zwischen den zu integrierenden Anwendungssystemen besteht eine n-zu-m-Beziehung.

Die Kommunikation zwischen den zu integrierenden Anwendungssystemen ist unidirektional.



Die Kommunikation zwischen den zu integrierenden Anwendungssystemen ist bidirektional.

Im Rahmen der Integrationslösung spielen mehrere Anwendungssysteme eine zentrale Rolle.



Im Rahmen der Integrationslösung spielt ein Anwendungssystem eine zentrale Rolle. Die Anbindung anderer Anwendungssysteme orientiert sich an diesem zentralen Anwendungssystem.

Für die Integration der Anwendungssysteme kann ein zentraler Hub eingesetzt werden.



Für die Integration der Anwendungssysteme kann kein zentraler Hub eingesetzt werden.

☐ Die Anzahl und die Komplexität der Schnittstellen

Die Anzahl der Schnittstellen ist gering.



Die Anzahl der Schnittstellen ist groß.

Die Komplexität einer Schnittstelle ist gering.



Die Komplexität einer Schnittstelle ist groß.

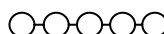
Die Anzahl an Datensätzen, die eine Schnittstelle passieren, ist gering.



Die Anzahl an Datensätzen, die eine Schnittstelle passieren, ist groß.

☐ Die Komplexität der Adapter

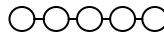
Die Komplexität eines Adapters ist gering.



Die Komplexität eines Adapters ist groß.

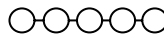
☐ **Die Anzahl der Formate der Kommunikation unter den zu integrierenden Anwendungssystemen**

Im Rahmen der Interaktion von zu integrierenden Anwendungssystemen sind wenige Systemgrenzen zu überwinden und Verarbeitungsschritte durchzuführen.



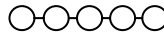
Im Rahmen der Interaktion von zu integrierenden Anwendungssystemen sind viele Protokoll- und Formatsprünge zu überwinden.

Die Anzahl der Datensatzformate ist gering.



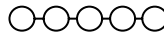
Die Anzahl der Datensatzformate ist groß.

Die Anzahl der Nachrichten-Formate ist gering.



Die Anzahl der Nachrichten-Formate ist groß.

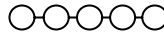
Die Integration aller Anwendungssysteme basiert auf den gleichen Kommunikationsarten und -formaten.



Die Integration der Anwendungssysteme basiert auf vielen unterschiedlichen Kommunikationsarten und -formaten.

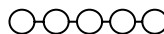
☐ **Die Anzahl und die Komplexität der Transformationen**

Die Anzahl der zu realisierenden Transformationen ("Mappings") ist gering.



Die Anzahl der zu realisierenden Transformationen ("Mappings") ist groß.

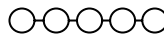
Die Komplexität einer zu realisierenden Transformation ("Mapping") ist gering.



Die Komplexität einer zu realisierenden Transformation ("Mapping") ist groß.

☐ **Die Komplexität der Datenstrukturen der Integrationslösung**

Die Komplexität des Datenmodells ist gering.



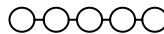
Die Komplexität des Datenmodells ist groß.

Die Anzahl der Datenbanktabellen ist gering.



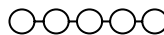
Die Anzahl der Datenbanktabellen ist groß.

Die Komplexität einer Datenbanktabelle ist gering.



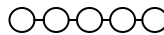
Die Komplexität einer Datenbanktabelle ist groß.

Die Anzahl der Datenelemente ist gering.



Die Anzahl der Datenelemente ist groß.

Die Komplexität eines Datenelements ist gering.



Die Komplexität eines Datenelements ist groß.

Die Informationsstrukturen sind flach.



Die Informationsstrukturen sind baumartig verschachtelt.

Die Komplexität der Struktur eines Datenobjekts ist gering.



Die Komplexität der Struktur eines Datenobjekts ist groß.

☐ **Der Einsatz eines kanonischen Datenmodells für die Durchführung von Transformationen**

Die Transformationen ("Mappings") werden anhand eines zentralen ("kanonischen") Datenmodells realisiert.



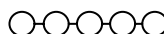
Die Transformationen ("Mappings") werden individuell zwischen den Anwendungssystemen realisiert.

Es kann ein einziges zentrales ("kanonisches") Datenmodell für die zu integrierenden Anwendungssysteme gefunden werden.



Mehrere zentrale ("kanonische") Datenmodelle, die jeweils für Teilmengen der zu integrierenden Anwendungssysteme gelten, müssen miteinander vereinbart werden.

Es muss kein zentrales ("kanonisches") Datenmodell geschaffen werden.



Es muss ein zentrales ("kanonisches") Datenmodell geschaffen werden.

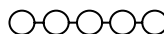
☐ **Die mit der Integrationslösung zu überwindenden Netzwerkgrenzen**

Die Integration der Anwendungssysteme erfolgt über wenige Netzwerkgrenzen hinweg.



Die Integration der Anwendungssysteme erfolgt über viele Netzwerkgrenzen hinweg.

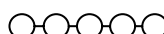
Die Integration der Anwendungssysteme erfolgt innerhalb einer demilitarisierten Zone/Sicherheitszone.



Die Integration der Anwendungssysteme erfolgt über mehrere demilitarisierten Zonen/Sicherheitszonen hinweg.

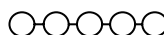
☐ **Die Gestaltung der Kommunikation zwischen den Anwendungssystemen**

Die Integration der Anwendungssysteme erfolgt auf der Basis synchroner Mechanismen ("request-reply").



Die Integration der Anwendungssysteme erfolgt auf der Basis asynchroner Mechanismen ("public-subscribe").

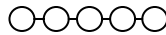
Die Kommunikation der zu integrierenden Anwendungssysteme basiert auf Bulk-Lieferungen.



Die Kommunikation der zu integrierenden Anwendungssysteme basiert auf Einzelsatzverarbeitung.

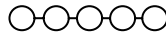
☐ **Das Ausmaß der Kopplung zwischen den Komponenten der Integrationslösung**

Die zu integrierenden Anwendungssysteme werden lose gekoppelt.



Die zu integrierenden Anwendungssysteme werden eng gekoppelt.

Bei der Gestaltung der Integrationslösung wird keine lose Kopplung zwischen den zu integrierenden Anwendungssystemen und den eingesetzten Integrationskomponenten angestrebt. Die Anwendungssysteme "kennen" die verwendete Technologie.



Bei der Gestaltung der Integrationslösung wird eine lose Kopplung zwischen den zu integrierenden Anwendungssystemen und den eingesetzten Integrationskomponenten angestrebt ("Transparenz der Technologie").

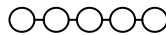
Die Integration der Anwendungssysteme erfolgt durch die Nutzung direkter Schnittstellen.



Die Integration der Anwendungssysteme erfolgt durch den Einsatz einer Middleware.

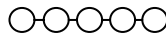
☐ **Die Realisierung der Steuerung der Interaktionen der zu integrierenden Anwendungssysteme**

Geschäftsprozesse werden nicht auf der Integrationsplattform abgebildet. Die Steuerung der Interaktionen zwischen den zu integrierenden Anwendungssystemen erfolgt durch die Anwendungssysteme selbst.



Geschäftsprozesse werden auf der Integrationsplattform abgebildet. Die Steuerung der Interaktionen zwischen den zu integrierenden Anwendungssystemen erfolgt über die Integrationsplattform.

Die in der Integrationslösung abzubildenden Geschäftsprozesse und Geschäftsregeln werden fest im Programmcode umgesetzt.



Die in der Integrationslösung abzubildenden Geschäftsprozesse und Geschäftsregeln werden extrahiert und beispielsweise als "business rules" implementiert.

☐ **Die Anzahl und die Komplexität der zu entwickelnden Module der Integrationslösung**

Die Anzahl der zu entwickelnden Module ist gering.



Die Anzahl der zu entwickelnden Module ist groß.

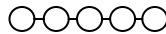
Die Komplexität eines zu entwickelnden Moduls ist gering.



Die Komplexität eines zu entwickelnden Moduls ist groß.

☐ **Die Anzahl und die Komplexität der zu realisierenden Funktionen**

Die Anzahl der zu realisierenden Funktionen ist gering.



Die Anzahl der zu realisierenden Funktionen ist groß.

Die Komplexität einer zu realisierenden Funktion ist gering.



Die Komplexität einer zu realisierenden Funktion ist groß.

☐ **Die Gestaltung der Authentifizierung der Benutzer**

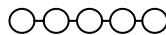
Eine Authentifizierung erfolgt mittels eines geheimen Schlüssels, d. h. ohne eine Passwortsynchronisation.



Eine Authentifizierung erfolgt über Benutzer- und Passwortsynchronisation.

☐ **Die Berücksichtigung der Fachlichkeit bei der Gestaltung der Integrationslösung**

Die Integrationslösung ist an die Geschäftsprozesse und die Betriebsführung der IT des Kunden angepasst.



Die Integrationslösung ist nicht an die Geschäftsprozesse und die Betriebsführung der IT des Kunden angepasst.

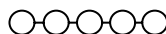
Das Zusammenspiel von Technologie und fachlichen Prozessen ist verstanden und kommuniziert.



Das Zusammenspiel von Technologie und fachlichen Prozessen ist nicht verstanden und kommuniziert.

☐ **Die Berücksichtigung der Gleichzeitigkeit von Geschäftsprozessen bei der Gestaltung der Integrationslösung**

Im Falle der Integration auf Datenebene und der Gleichzeitigkeit von Geschäftsprozessen wird der Status vor der Persistierung von Daten geprüft und plausibilisiert.

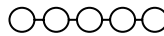


Im Falle der Integration auf Datenebene und der Gleichzeitigkeit von Geschäftsprozessen wird der Status vor der Persistierung von Daten nicht geprüft und plausibilisiert.

4. Die eingesetzten Integrationsprodukte, Technologien und Entwicklungswerkzeuge

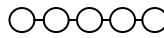
☐ Die Verfügbarkeit von Integrationsprodukten

Die Integrationslösung basiert auf standardisierten Integrationsprodukten.



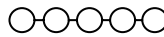
Die Integrationslösung wird vollständig eigenentwickelt.

Die Integrationslösung basiert auf konfigurierbaren Integrationsprodukten.



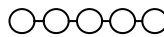
Die Integrationslösung wird mit Entwicklungswerkzeugen umgesetzt, die keine fertige Lösung produzieren.

Moderne Integrationsprodukte sind verfügbar und geben die Integrationslösung vor.



Es mangelt an Unterstützung der Integrationsaufgaben durch Integrationsprodukte.

Ein Standardadapter steht zur Verfügung.



Ein Adapter muss selber entwickelt werden.

☐ Die Schwierigkeit des Verstehens und Verwendens der eingesetzten Integrationsprodukte

Das eingesetzte Integrationsprodukt basiert auf generischen Standards.



Das eingesetzte Integrationsprodukt ist proprietär und sehr speziell.

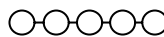
Das eingesetzte Integrationsprodukt kann sehr leicht installiert/aufgesetzt werden.



Das eingesetzte Integrationsprodukt ist schwer zu installieren/aufzusetzen.

☐ Die Qualität der eingesetzten Integrationsprodukte

Der Reifegrad des eingesetzten Integrationsproduktes ist hoch.



Der Reifegrad des eingesetzten Integrationsproduktes ist niedrig.

Das eingesetzte Integrationsprodukt ist etabliert und gut dokumentiert.



Das eingesetzte Integrationsprodukt ist nicht etabliert und wenig dokumentiert.

Das eingesetzte Integrationsprodukt ist stabil und zuverlässig.



Das eingesetzte Integrationsprodukt ist instabil und unberechenbar.

Das eingesetzte Integrationsprodukt kann standardmäßig in der Integrationslösung eingesetzt werden.



Die Standardschnittstellen des eingesetzten Integrationsproduktes sind fehlerhaft.

☐ Die Änderungsdynamik der eingesetzten Integrationsprodukte

Es sind keine Updates des eingesetzten Integrationsproduktes erforderlich.



Das eingesetzte Integrationsprodukt weist kurze Updateintervalle auf.

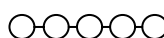
Es ist kein Versionswechsel des eingesetzten Integrationsproduktes während der Projektlaufzeit erforderlich.



Es ist ein Versionswechsel des eingesetzten Integrationsproduktes während der Projektlaufzeit erforderlich.

☐ Die Eignung der eingesetzten Integrationsprodukte für die Integrationslösung

Es werden geeignete Integrationsprodukte in der Integrationslösung eingesetzt.



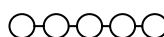
Es werden ungeeignete Integrationsprodukte in der Integrationslösung eingesetzt.

Das eingesetzte Integrationsprodukt erfüllt die gestellten funktions- und qualitätsbezogenen Anforderungen.



Das eingesetzte Integrationsprodukt erfüllt die gestellten funktions- und qualitätsbezogenen Anforderungen nicht.

Es ist keine funktionale Erweiterung des eingesetzten Integrationsproduktes notwendig.



Es werden funktionale Erweiterungen des eingesetzten Integrationsproduktes durch dessen Hersteller erforderlich.

Das eingesetzte Integrationsprodukt bietet tatsächlich die von dem Hersteller genannten Features.



Das eingesetzte Integrationsprodukt bietet nicht die vom Hersteller genannten Features.

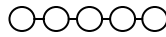
Das eingesetzte Integrationsprodukt unterstützt standardmäßig ein erforderliches Nachrichtenformat oder eine anzubindende Schnittstelle.



Das eingesetzte Integrationsprodukt unterstützt ein erforderliches Nachrichtenformat oder eine anzubindende Schnittstelle nicht standardmäßig.

☐ **Die Abhängigkeit von den Herstellern der eingesetzten Integrationsprodukte**

Es gibt ein starkes Lieferantenmanagement bezüglich der eingesetzten Integrationsprodukte.



Es gibt ein schwaches Lieferantenmanagement bezüglich der eingesetzten Integrationsprodukte. Man gerät gegenüber Lieferanten in Abhängigkeiten.

Das eingesetzte Integrationsprodukt stammt aus einer Hand.



Bei dem eingesetzten Integrationsprodukt bestehen Abhängigkeiten von Drittherstellern.

☐ **Die Unterstützung durch die Hersteller der eingesetzten Integrationsprodukte**

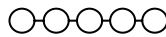
Der Hersteller des eingesetzten Integrationsproduktes ist etabliert und hat entsprechende Supportmöglichkeiten.



Der Hersteller des eingesetzten Integrationsproduktes ist klein und hat nur geringe Supportmöglichkeiten.

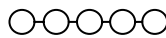
☐ **Die eingesetzten Technologien**

Die Anzahl der eingesetzten Technologien ist niedrig.



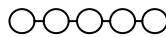
Die Anzahl der eingesetzten Technologien ist groß.

Die eingesetzte Technologie lässt sich durch die Projektmitarbeiter leicht anwenden.



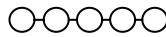
Die eingesetzte Technologie lässt sich durch die Projektmitarbeiter schwer anwenden.

Die eingesetzte Technologie ist etabliert.



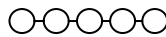
Die eingesetzte Technologie ist neuartig.

Die eingesetzte Technologie bietet Unterstützung durch Standardbibliotheken.



Man muss vollständig eigenentwickeln.

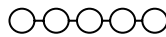
Die Integrationslösung basiert auf programmatischer Technologie (d. h. Realisierung mittels Programmiersprache).



Die Integrationslösung basiert auf deklarativer Technologie (d. h. Realisierung mittels Auszeichnungssprachen, wie XML).

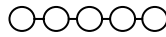
☐ **Die Unterstützung durch Entwicklungswerkzeuge**

Es wird ein vorhandenes Entwicklungswerkzeug eingesetzt.



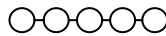
Es wird ein integriertes Entwicklungswerkzeug für die Integrationslösung entwickelt.

Moderne Entwicklungswerkzeuge sind verfügbar.



Moderne Entwicklungswerkzeuge sind nicht verfügbar.

Die Services einer serviceorientierten Architektur werden aufgrund eines generischen Ansatzes durch ein Entwicklungswerkzeug generiert.



Die Services einer serviceorientierten Architektur werden jeweils eigenentwickelt.

☐ **Die Einhaltung der Standards der Anwendungssysteme**

Die Integration eines ERP-Systems erfolgt unter Einhaltung dessen Standards.

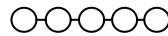


Die Integration eines ERP-Systems erfolgt durch Eigenentwicklung bei Abweichung vom Standard des ERP-Systems.

5. Das Testen der Integrationslösung und die Überleitung in die Nutzung

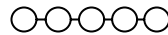
☐ Der Umfang des Testens

Es werden wenige Teststufen durchgeführt.



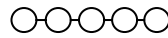
Es werden viele Teststufen durchgeführt.

Es werden wenige Regressionstests durchgeführt.



Es werden viele Regressionstests durchgeführt.

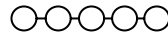
Es werden wenige Testfälle durchgeführt.



Es werden viele Testfälle durchgeführt.

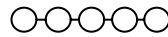
☐ Die Sorgfalt des Testens

Es gibt eine hohe Testfallüberdeckung bezüglich der Schnittstellen. Es gibt eine lange Testphase.



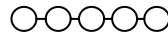
Es gibt eine geringe Testfallüberdeckung bezüglich der Schnittstellen. Es gibt eine kurze Testphase.

Es wird sehr umfangreich und systematisch getestet.



Es wird nicht sehr umfangreich und systematisch getestet.

Es gibt eine klare zeitliche Einplanung der Integrationstestphase.



Es gibt keine klare zeitliche Einplanung der Integrationstestphase.

Es gibt ein definiertes Testvorgehen.



Es gibt kein definiertes Testvorgehen.

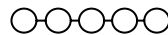
☐ Die herrschenden Umstände zur Durchführung des Testens

Eine komplexe Testumgebung muss nicht aufgebaut werden.



Eine komplexe Testumgebung muss aufgebaut werden.

Testsysteme der zu integrierenden Anwendungssysteme sind vorhanden.



Die zu integrierenden Anwendungssysteme müssen für Entwicklung und Test simuliert werden.

Testdaten für die zu integrierenden Anwendungssysteme sind vorhanden.



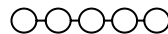
Testdaten für die zu integrierenden Anwendungssysteme sind nicht vorhanden.

Die Tests sind vollständig automatisiert.



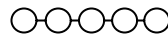
Die Tests sind vollständig manuell.

Erforderliche Korrekturen können unmittelbar vorgenommen werden. Entwicklung und Test durchdringen einander.



Es gibt lange Verzögerungszeiten ("turn-around-time") um Korrekturen vornehmen zu können.

Die eingesetzte Technologie ist Interpreter-basiert. Man kann im laufenden Betrieb Änderungen vornehmen und sofort testen.



Die eingesetzte Technologie ist compiler-basiert. Man muss nach einer Änderung Testdurchläufe starten.

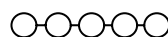
☐ Die Verteilung der Zuständigkeiten beim Testen

Die Zuständigkeit für den Test der gesamten Integrationslösung ist gebündelt.



Es gibt getrennte Zuständigkeiten für den Test von Teilen der Integrationslösung.

Es gibt ein internes und vollständig verstandenes Testsystem.



Es gibt verteilte Testsysteme, die getrennt betreut werden.

☐ Das Vorgehen bei der Überleitung in die Nutzung

Die Anzahl der durchzuführenden Trainings ist klein.



Die Anzahl der durchzuführenden Trainings ist groß.

Der Auslieferungsprozess ist einfach.

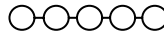


Der Auslieferungsprozess ist komplex.

6. Der Kunde und die Stakeholder der Integrationslösung

☐ Die Beziehung zum Kunden

Man führt das Integrationsvorhaben als externer Dienstleister für einen Kunden durch.



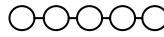
Man führt das Integrationsvorhaben als interner Dienstleister durch.

Es gibt einen klaren Arbeitsauftrag und eine klare Trennung von Auftraggeber und Auftragnehmer.



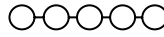
Es gibt keinen klaren Arbeitsauftrag und der Kunde ist in die Entwicklung der Integrationslösung involviert.

Man steht seit langem in Beziehung zum Kunden des Integrationsvorhabens.



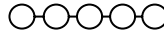
Das Integrationsvorhaben stellt eine erstmalige oder einmalige Beziehung zum Kunden dar.

Die Integrationslösung wird lediglich entwickelt und bereitgestellt.



Die Projektmitarbeiter leisten Unterstützung ("Support") für die Betriebsseite.

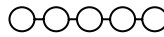
Der Kunde hat Erfahrung darin, Entwicklungsvorhaben gemeinsam mit externen Dienstleistern durchzuführen.



Der Kunde kauft IT-Lösungen als Black-Box ("wie einen PC").

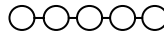
☐ Die Homogenität der Stakeholder des Integrationsvorhabens

Die Anzahl der beteiligten Parteien ist gering.



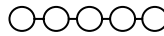
Die Anzahl der beteiligten Parteien ist groß.

Das Integrationsvorhaben betrifft Anwendungssysteme nur einer organisatorischen Einheit.



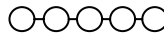
Das Integrationsvorhaben betrifft Anwendungssysteme mehrerer organisatorischer Einheiten.

Das Integrationsvorhaben betrifft Anwendungssysteme eines Unternehmens.



Das Integrationsvorhaben betrifft Anwendungssysteme mehrerer Unternehmen.

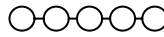
Das Integrationsvorhaben betrifft nur Anwendungssysteme des Kunden.



Das Integrationsvorhaben betrifft Anwendungssysteme des Kunden sowie Anwendungssysteme externer Partner des Kunden.

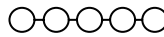
☐ Die Einstellung der Stakeholder zum Integrationsvorhaben

Die Stakeholder des Integrationsvorhabens haben eine gemeinsame Vision.



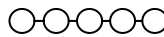
Die Stakeholder des Integrationsvorhabens haben unterschiedliche Zielsetzungen.

Das Integrationsvorhaben wird durch die Benutzer der zu integrierenden Systeme unterstützt.



Es gibt eine Abwehrhaltung der Benutzer der zu integrierenden Systeme gegen das Integrationsvorhaben.

Die Stakeholder des Integrationsvorhabens haben keine Angst vor damit verbundenen Veränderungen.



Die Stakeholder des Integrationsvorhabens haben Angst vor damit verbundenen Veränderungen.

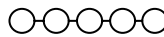
Die zu entwickelnde Integrationslösung ist vollständig neuartig.



Die zu entwickelnde Integrationslösung konkurriert gegen bestehende Lösungen beim Kunden.

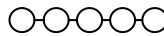
☐ Die Ausprägung der Kommunikation mit dem Kunden

Die Anforderungen können unmittelbar mit dem Endkunden abgestimmt werden.



Die Anforderungen des Endkunden werden über Dritte kommuniziert.

Es besteht ausschließlich ein formalisierter Kontakt zwischen den Entwicklern und den Fachabteilungen.



Die Fachabteilungen können die Entwickler direkt kontaktieren.

Es gibt eine starke und formale Einbeziehung der Fachabteilungen.



Die Einbeziehung der Fachabteilungen erfolgt spontan.

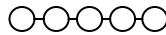
Es erfolgt eine fortwährende Kommunikation von Zwischenergebnissen gegenüber dem Kunden.



Dem Kunden werden keine Zwischenergebnisse kommuniziert.

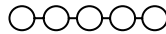
☐ **Das Ausmaß der Unterstützung durch den Kunden**

Es gibt eine starke Kundenbeteiligung und -mitarbeit.



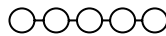
Es gibt eine sehr geringe Kundenbeteiligung und -mitarbeit.

Der Kunde liefert fachliche Unterstützung bei der Realisierung.



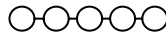
Der Kunde ist nicht in die Realisierung eingebunden.

Der Kunde kann seine Mitwirkungspflicht erfüllen.



Der Kunde kann seine Mitwirkungspflicht nicht erfüllen.

Die Projektmitarbeiter werden durch die Endanwender unterstützt.



Die Projektmitarbeiter werden nicht durch die Endanwender unterstützt.

Die Management Attention für das Integrationsvorhaben ist hoch.



Die Management Attention für das Integrationsvorhaben ist niedrig.

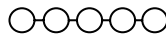
☐ **Das Ausmaß von Erfahrung und Wissen des Kunden**

Der Kunde hat Erfahrung mit der Anwendungsintegration und verfügt über ein klares und einheitliches Begriffssystem zu deren Beschreibung.



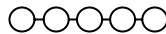
Der Kunde hat keine Erfahrung mit der Anwendungsintegration und verfügt über kein klares und einheitliches Begriffssystem zu deren Beschreibung.

Der Kunde hat sich mit der Anwendungsintegration beschäftigt und kennt sich damit aus.



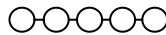
Der Kunde muss bezüglich der Anwendungsintegration "missioniert" werden.

Der Erkenntnisgrad des Managements bezüglich der Probleme von Integrationsvorhaben ist hoch.



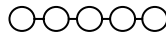
Der Erkenntnisgrad des Managements bezüglich der Probleme von Integrationsvorhaben ist gering.

Die Business-Stakeholder besitzen ein hohes IT-Verständnis.



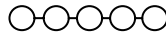
Die Business-Stakeholder besitzen kein IT-Verständnis.

Der Kunde ist mit den zu integrierenden Anwendungssystemen vertraut.



Der Kunde ist mit den zu integrierenden Anwendungssystemen nicht vertraut.

Der Kunde ist fachlich und technisch sehr kompetent.



Der Kunde ist fachlich und technisch eingeschränkt kompetent.

☐ **Die Gestaltung der Zuständigkeiten auf Kundenseite**

Es gibt einen verantwortlichen Ansprechpartner auf Kundenseite.



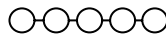
Es gibt viele verschiedene Ansprechpartner auf Kundenseite.

Man hat unmittelbaren Kontakt zu den Entscheidungsträgern des Kunden.



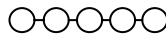
Die Ansprechpartner beim Kunden haben keine Entscheidungsbefugnis.

Im Laufe des Integrationsvorhabens ist keine weitere Entscheidung des Kunden notwendig.



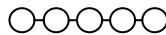
Der Kunde hat keine klaren und kompakten Entscheidungsstrukturen. Keiner der Ansprechpartner traut sich eine Entscheidung zu treffen.

Es existiert eine etablierte, mehrheitlich demokratische Entscheidungsstruktur beim Kunden.



Es existiert eine autokratische Entscheidungsstruktur beim Kunden.

Beim Kunden gibt es eine zentrale Leitung der IT.



Beim Kunden erfolgt die Leitung der IT dezentral.

Beim Kunden gibt es eine starke Leitung der IT.

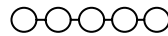


Beim Kunden gibt es eine schwache Leitung der IT.

7. Die Projektmitarbeiter

☐ Die Anzahl der Projektmitarbeiter

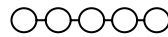
Es gibt wenige Projektmitarbeiter.



Es gibt viele Projektmitarbeiter.

☐ Die allgemeine Produktivität der Projektmitarbeiter

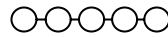
Die Produktivität der Projektmitarbeiter ist hoch.



Die Produktivität der Projektmitarbeiter ist niedrig.

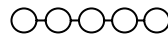
☐ Die Homogenität der Projektmitarbeiter

Die Projektmitarbeiter bilden ein homogenes Team mit einem gemeinsamen Verständnis von Begriffen und Konzepten.



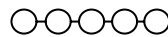
Die Projektmitarbeiter bilden lose Teamverbände ohne ein gemeinsames Verständnis von Begriffen und Konzepten.

Die Projektmitarbeiter entstammen dem selben Unternehmen und bilden ein homogenes Team.



Die Projektmitarbeiter entstammen unterschiedlichen Unternehmen und bilden ein heterogenes Team.

Die Projektmitarbeiter formen ein kleines Spezialistenteam.



Die Projektmitarbeiter formen sehr heterogene Teamstrukturen.

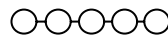
☐ Erfahrung und Wissen der Projektmitarbeiter

Es ist kein tiefgehendes technisches Spezialwissen erforderlich.



Punktuell ist ein tiefgehendes technisches Spezialwissen erforderlich.

Kompetente Projektmitarbeiter sind verfügbar.



Kompetente Projektmitarbeiter sind nicht verfügbar.

Personen mit Spezialisten-Know-how sind verfügbar.



Personen mit Spezialisten-Know-how sind nicht verfügbar.

Die Projektmitarbeiter haben sehr gutes Fachwissen.



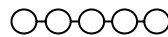
Die Projektmitarbeiter haben kein Fachwissen.

Die Projektmitarbeiter sind mit den zu integrierenden Anwendungssystemen vertraut.



Die Projektmitarbeiter sind nicht mit den zu integrierenden Anwendungssystemen vertraut.

Die Projektmitarbeiter sind erfahren.



Die Projektmitarbeiter sind unerfahren.

Die mit der Analyse betrauten Projektmitarbeiter haben sehr gute Software Engineering Skills ("hard skills").



Die mit der Analyse betrauten Projektmitarbeiter haben keine Software Engineering Skills ("hard skills").

Die Projektmitarbeiter sind mit der eingesetzten Technologie vertraut.



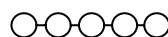
Die Projektmitarbeiter sind mit der eingesetzten Technologie nicht vertraut.

Die Projektmitarbeiter sind Experten im Umgang mit dem eingesetzten Integrationsprodukt.



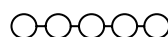
Die Projektmitarbeiter wissen nicht, welche Features des eingesetzten Integrationsproduktes in der Problemdomäne eingesetzt werden können.

Die Anwendungssysteme und Geschäftsprozesse sind den mit dem Testen betrauten Projektmitarbeitern bekannt.



Die Anwendungssysteme und Geschäftsprozesse sind den mit dem Testen betrauten Projektmitarbeitern nicht bekannt.

Die Projektbeteiligten besitzen von Beginn an das erforderliche Wissen.



Die Projektbeteiligten müssen das erforderliche Wissen während der Projektlaufzeit aufbauen.

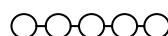
☐ Die Qualität der Zusammenarbeit unter den Projektmitarbeitern

Die Projektmitarbeiter haben eine "Open Source Philosophie" gegenüber anderen Projektmitarbeitern.



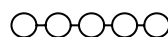
Die Projektmitarbeiter gewähren anderen Projektmitarbeitern keinen Einblick in den von ihnen entwickelten Programmcode.

Unter den Projektbeteiligten herrscht Teamgeist.



Unter den Projektbeteiligten herrscht kein Teamgeist.

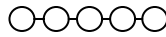
Die mit der Analyse betrauten Projektmitarbeiter besitzen sehr hohe soziale Kompetenz ("soft skills").



Die mit der Analyse betrauten Projektmitarbeiter sind nicht teamfähig.

☐ **Die Kontinuität der Projektmitarbeiter**

Das Team der Projektmitarbeiter besteht seit längerem.



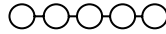
Das Team der Projektmitarbeiter muss wiederholt aus verschiedenen Quellen zusammengestellt werden.

Die Projektmitarbeiter formen ein eingespieltes Team.



Die Projektmitarbeiter sind nicht miteinander vertraut.

Es gibt dedizierte Mitarbeiter für das Integrationsvorhaben.



Die Mitarbeiter arbeiten parallel auch an projektexternen Aufgaben. Die Ansprechpartner für das Integrationsvorhaben wechseln.

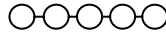
☐ **Die Einstellung der Projektmitarbeiter zum Integrationsvorhaben**

Die Projektmitarbeiter sind motiviert.



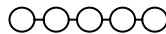
Die Projektmitarbeiter sind nicht motiviert.

Die Projektmitarbeiter lösen sich von persönlichen Vorlieben für bestimmte Technologien.



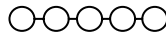
Die Projektmitarbeiter wollen nur in der ihnen vertrauten Art unter Einsatz bestimmter Technologien entwickeln.

Die Projektmitarbeiter sind bereit, sich in neue Systeme/Technologien einzuarbeiten.



Die Projektmitarbeiter sind nicht bereit sich in neue Systeme/Technologien einzuarbeiten.

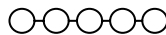
Die Projektmitarbeiter haben eine gesamtsystemorientierte Perspektive.



Die Projektmitarbeiter haben eine auf Anwendungssysteme fokussierte Perspektive.

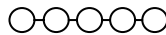
☐ **Die Zusammenarbeit mit weiteren Unternehmen bei der Entwicklung der Integrationslösung**

Die Realisierung der Integrationslösung wird intern durchgeführt.



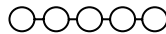
Die Realisierung der Integrationslösung wird an externe Dienstleister vergeben.

Die Entwicklung der Integrationslösung wird selbstständig durchgeführt.



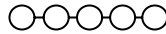
Die Entwicklung der Integrationslösung erfolgt gemeinsam mit weiteren externen Dienstleistern.

Ein Teil der Verantwortung kann an einen erfahrenen externen Partner abgegeben werden.



Die Gesamtverantwortung liegt in der eigenen Hand.

Die Entwicklung erfolgt in Zusammenarbeit mit einem Unternehmen, mit dem ein partnerschaftliches Verhältnis besteht.



Die Entwicklung erfolgt in Zusammenarbeit mit einem Unternehmen, mit dem ein kommerzielles Verhältnis besteht.

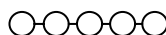
☐ **Die Qualität der allgemeinen Kommunikation unter den Projektbeteiligten**

Die Schlüsselpersonen des Integrationsvorhabens haben eine gute Beziehung zueinander und kommunizieren viel.



Die Schlüsselpersonen des Integrationsvorhabens haben eine schlechte Beziehung zueinander und kommunizieren wenig.

Die Kommunikation zwischen allen Projektbeteiligten ist gewährleistet.



Die einzelnen Projektbeteiligten arbeiten isoliert voneinander jeweils nur an ihren eigenen Arbeitspaketen.

8. Das Management des Integrationsvorhabens

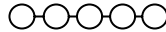
☐ Die Qualität des Managements des Integrationsvorhabens

Es gibt ein starkes Projektmanagement.



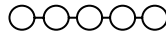
Es gibt kein starkes Projektmanagement.

Es gibt ein Projektmanagement mit klarem Scope und klar abgegrenzter Kompetenz.



Es gibt direkte Interventionen des Linienmanagements in das Projekt.

Das Projektmanagement besitzt hohe soziale Kompetenz und fachliches Know-how.



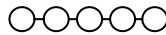
Das Projektmanagement besitzt keine soziale Kompetenz und fachliches Know-how.

Der Projektleiter ist sensibilisiert für Integrationsaspekte.



Der Projektleiter ist nicht sensibilisiert für Integrationsaspekte.

Im Projekt ist ein Risikomanagement implementiert.



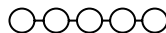
Im Projekt ist kein Risikomanagement implementiert.

Es ist von Anfang an ein Changemanagement aufgesetzt.



Es gibt kein Changemanagement.

Das Team der Projektmitarbeiter wird kontrolliert von oben nach unten aufgebaut.



Das Team der Projektmitarbeiter wird zu schnell aufgebaut.

Die Planung des Projektes erfolgt anhand eines Vorgehensmodells.



Die Planung des Projektes erfolgt spontan und durch den Endtermin getrieben.

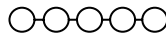
Das Integrationsvorhaben wird in mehrere Meilensteine eingeteilt und diese Planung wird befolgt.



Das Integrationsvorhaben weist sehr lange Teilprojektphasen auf.

☐ Das Ausmaß von vorausschauendem Vorgehen

Das Integrationsvorhaben wurde gut vorbereitet.



Das Integrationsvorhaben wurde unzureichend vorbereitet.

Das Integrationsvorhaben folgt dem Grundsatz "erst denken, dann handeln".



In dem Integrationsvorhaben "sind Denken und Handeln eins".

☐ Die Gestaltung der Aufbauorganisation des Integrationsvorhabens

Zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter bestehen keine Interdependenzen.



Zwischen den Aufgaben einzelner Gruppen der Projektmitarbeiter bestehen starke Interdependenzen.

Es gibt klare Organisationsschnittstellen in der Projektorganisation.



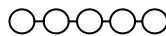
Es gibt eine netzartige Kommunikationsstruktur in der Projektorganisation.

Die Abstimmung und die Kommunikation zwischen Entwicklung und Test werden gefördert und durch dedizierte Projektmitarbeiter ("Integrationsteam") sichergestellt.



Die Abstimmung und die Kommunikation zwischen Entwicklung und Test werden nicht gefördert. Zwischen Entwicklung und Test besteht ein Nadelöhr.

Die Projektorganisation ist teamorientiert.



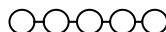
Die Projektorganisation ist hierarchisch.

Es gibt klar definierte Verantwortlichkeiten hinsichtlich der Integrationsfunktionalität.



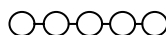
Es gibt eine Vermischung der Verantwortlichkeiten hinsichtlich der Integrationsfunktionalität und fachlicher Funktionalität.

Die Verantwortlichkeiten der Projektbeteiligten sind klar definiert. Die organisatorische Gestaltung des Integrationsvorhabens ist projektorientiert, bei Ausrichtung an den zu liefernden Ergebnissen.



Die Verantwortlichkeiten der Projektbeteiligten sind unklar. Das Integrationsvorhaben wird aus der Linienorganisation des Unternehmens heraus durchgeführt. Die Ergebnisse der einzelnen Linien müssen kombiniert werden.

Es gibt ein zentrales Mandat für die Modellierung der zu unterstützenden Geschäftsprozesse.



Das Mandat für die Modellierung der zu unterstützenden Geschäftsprozesse ist auf mehrere Teilprojekte verteilt.

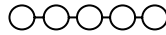
☐ **Der verfolgte Entwicklungsansatz**

Der Kunde will zu einem bestimmten Termin einen definierten Lieferumfang haben. Es wird ein sequenzieller/dokumentenbasierter Entwicklungsansatz verfolgt.



Der Kunde will zu einem bestimmten Termin einen definierten Lieferumfang haben. Es wird ein evolutionärer Entwicklungsansatz verfolgt.

Der Kunde ist terminorientiert und in der Gestaltung der Funktionalität flexibler. Es wird ein evolutionärer Entwicklungsansatz verfolgt.



Der Kunde ist terminorientiert und in der Gestaltung der Funktionalität flexibler. Es wird ein sequenzieller/dokumentenbasierter Entwicklungsansatz verfolgt.

Unter den Projektbeteiligten herrscht ein Konsens über die verfolgte Entwicklungsmethode.



Unter den Projektbeteiligten herrscht kein Minimumkonsens über die verfolgte Entwicklungsmethode.

Die Releasezyklen der zu integrierenden Anwendungssysteme und die Vorgehensmodelle, auf denen ihre Entwicklung beruht, können im Integrationsvorhaben berücksichtigt werden.



Die zu integrierenden Anwendungssysteme besitzen unterschiedliche Releasezyklen und ihre Entwicklung beruht auf unterschiedlichen Vorgehensmodellen.

9. Die Rahmenbedingungen des Integrationsvorhabens

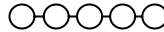
☐ Die Dauer des Integrationsvorhabens

Das Integrationsvorhaben hat eine kurze Laufzeit.



Das Integrationsvorhaben hat eine lange Laufzeit.

Das Integrationsvorhaben umfasst wenige Releases.



Das Integrationsvorhaben umfasst viele Releases.

☐ Die Neuartigkeit des Integrationsvorhabens

Das Integrationsvorhaben stellt die Fortführung eines vorangehenden Projektes dar.



Das Integrationsvorhaben ist vollständig neu.

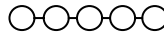
☐ Die Klarheit der Zielsetzung des Integrationsvorhabens

Das Integrationsvorhaben ist auf die Bedürfnisse des Einzelfalls ausgerichtet.



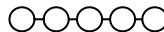
Das Integrationsvorhaben soll als Vorbild/Vorreiter für ein verändertes Vorgehen in der IT dienen.

Das Integrationsvorhaben verfolgt klar abgegrenzte Zielvorgaben.



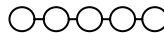
Das Integrationsvorhaben stellt einen ersten Schritt auf einem langen Weg dar.

Die Entwicklung einer umfassenden Integrationsarchitektur wurde explizit als Infrastrukturprojekt aufgesetzt.



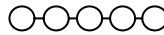
Das Vorhaben, eine umfassende Integrationsarchitektur zu entwickeln, ist historisch gewachsen.

Neben der Anwendungsintegration sind wenige begleitenden Maßnahmen durchzuführen.



Neben der Anwendungsintegration sind viele begleitenden Maßnahmen durchzuführen.

Mit der Anwendungsintegration werden Kostenvorteile auf hoher Budgetebene angestrebt.



Das Integrationsvorhaben ist charakterisiert durch ein permanentes Streben nach Kostenvorteilen auch im Kleinen.

☐ Die Bedeutung des Integrationsvorhabens für den Kunden

Die Anwendungsintegration hat lediglich interne Auswirkungen für die beteiligten Unternehmen.



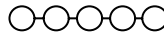
Die Anwendungsintegration hat eine Außenwirkung im Hinblick auf den Endkunden.

Ein Misserfolg des Integrationsvorhabens ist für den Kunden nicht geschäftsfährdend.



Das Integrationsvorhaben ist erfolgskritisch für den Kunden.

Die zu integrierenden Anwendungssysteme sind nicht geschäftskritisch.



Die zu integrierenden Anwendungssysteme sind geschäftskritisch.

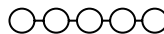
☐ Das Ausmaß der Internationalität des Integrationsvorhabens

Das Integrationsvorhaben ist national.



Das Integrationsvorhaben ist international.

Die Dokumentation der Integrationslösung erfolgt in einer Sprache.



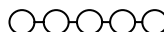
Die Dokumentation der Integrationslösung erfolgt in mehreren Sprachen.

Die Projektbeteiligten sind mit der Projektsprache sehr vertraut.



Die Projektbeteiligten sind mit der Projektsprache nicht vertraut.

Die Projektbeteiligten entstammen einem gemeinsamen Sprach- und Kulturraum.



Die Projektbeteiligten entstammen unterschiedlichen Sprach- und Kulturräumen.

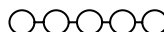
☐ Die räumliche Verteilung der Projektbeteiligten

Die Projektbeteiligten befinden sich an einem Standort.



Die Projektbeteiligten sind räumlich getrennt.

Die Projektbeteiligten befinden sich in derselben Zeitzone.



Die Projektbeteiligten befinden sich in unterschiedlichen Zeitzonen.

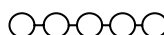
☐ Die verfügbare Infrastruktur zur Durchführung des Integrationsvorhabens

Es gibt angemessene getrennte Umgebungen für Entwicklung, Test und Abnahme.



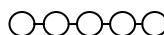
Die Entwicklung der Integrationslösung erfolgt unmittelbar in der Produktivumgebung.

Die Produktivumgebung kann in Entwicklungs- und Testumgebungen nachgebildet werden.



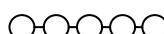
Die Produktivumgebung kann nicht nachgebildet werden. Entwicklungs- und Testumgebungen entsprechen nicht der Produktivumgebung.

Die Integrationslösung muss lediglich in der Produktivumgebung eingeführt ("deployed") werden.



Die Integrationslösung muss in Entwicklungs-, Test- und Produktivumgebungen eingeführt ("deployed") werden.

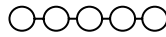
Die räumliche und technische Ausstattung der Projektmitarbeiter ist optimal.



Es ist keine angemessene räumliche und technische Infrastruktur für die Projektmitarbeiter vorhanden.

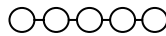
☐ Die Abhängigkeit des Integrationsvorhabens von anderen Projekten

Bei dem Integrationsvorhaben handelt es sich um ein eigenständiges Projekt.



Das Integrationsvorhaben ist ein Teilaspekt eines größeren, umfassenden Projektes.

Es gibt keine Parallelprojekte von denen man abhängig ist.



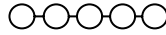
Es gibt Parallelprojekte von denen man abhängig ist.

Es ist keine Abstimmung mit anderen Projekten erforderlich.



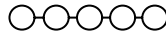
Es ist eine Abstimmung mit anderen Projekten erforderlich.

Man kann die Releaseplanung für das Integrationsvorhaben eigenständig vornehmen.



Das Integrationsvorhaben muss in eine bestehende Releaseplanung eingebettet werden.

Das Integrationsvorhaben ist ein Teilaspekt eines Projektes mit kurzer Gesamtdauer bzw. geringem Gesamtvolumen.



Das Integrationsvorhaben ist ein Teilaspekt eines Projektes mit langer Gesamtdauer bzw. großem Gesamtvolumen.

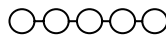
☐ Die bei der Entwicklung der Integrationslösung zu berücksichtigenden Vorgaben

Eine beim Kunden bereits eingesetzte Integrationsplattform ist für die Integrationslösung vorgegeben.



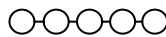
Man hat Wahlfreiheit bezüglich eines neu einzuführenden Integrationsproduktes.

Man hat Wahlfreiheit bezüglich der einzusetzenden Integrationsprodukte und der verwendeten Schnittstellen.



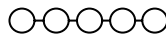
Es gibt Vorgaben bezüglich der einzusetzenden Integrationsprodukte oder Schnittstellen, die für die Lösung des Integrationsproblems nicht geeignet oder notwendig sind.

Es gibt klare Vorgaben des Kunden für die Entwicklung ("starke Governance" beim Kunden).



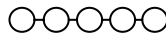
Die Entwicklung wird nicht durch den Kunden kontrolliert ("keine starke Governance" beim Kunden).

Es gibt keine bestimmten Anforderungen des Kunden, wie eine Software zu entwickeln ist.



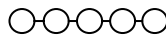
Es gibt bestimmte Anforderungen des Kunden, wie eine Software zu entwickeln ist.

Der Kunde mischt sich nicht in die Entwicklung ein.



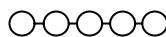
Der Kunde mischt sich in die Entwicklung ein.

Das Integrationsvorhaben bezweckt die einmalige Verbindung der zu integrierenden Anwendungssysteme.



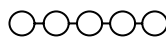
Der Lebenszyklus der zu integrierenden Anwendungssysteme muss bei der Entwicklung der Integrationslösung berücksichtigt werden. Systematische Veränderungen der zu integrierenden Anwendungssysteme (insbesondere der Schnittstellen) müssen unterstützt werden

Die Integrationslösung wird aufgrund eigener Konzepte maßgeschneidert.



Die Integrationslösung basiert auf einem standardisierten Paradigma, dessen Umsetzung einen Overhead bedeutet.

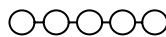
Datenschutzrechtliche Bestimmungen sind von geringer Bedeutung.



Datenschutzrechtliche Bestimmungen sind von großer Bedeutung.

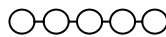
☐ Das verfügbare Budget / die zulässige Entwicklungsdauer

Das Budget des Kunden ist knapp. Der Kunde hat beschränkte finanzielle Mittel.



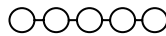
Das Budget des Kunden ist ausreichend. Der Kunde hat große finanzielle Mittel.

Man kann bedächtig entwickeln.



Es herrscht großer Zeitdruck.

Es gibt zwar einen vorgegebenen Plan für den Projektverlauf aber man kann Abweichungen von den Meilensteinen einkalkulieren.



Es gibt vorgegebene Releasetermine, die nicht verändert werden können.